NADC 77157-30

NOR 76-190

WAVE DRAG REDUCTION FOR AIRCRAFT FUSELAGE - WING CONFIGURATIONS

VOLUME II: Manual for Computer Programs

C.W. Chu, J. Der, Jr., H. Ziegler

Northrop Corporation Aircraft Division

FINAL REPORT OCTOBER 1977

Contract N62269-75-C-0537

NAVAL AIR DEVELOPMENT CENTER
Warminster, Pennsylvania 18974

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) NADC-77157-30 - (19) Vol-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) WAVE DRAG REDUCTION FOR AIRCRAFT FUSELAGE-WING CONFIGURATIONS. Volume II. Manual for Computer Programs.		5. ~~TYPE OF REPORT & PERIOD COVERED~~ FINAL rept. 30 June 1975-30 Oct. 1976
		6. PERFORMING ORG. REPORT NUMBER (14) NOR-76-190-Vol-2
7. AUTHOR(s) (10) C. W. Chu, J. Der, Jr., H. Ziegler		8. CONTRACT OR GRANT NUMBER(s) (15) N62269-75-C-0537
9. PERFORMING ORGANIZATION NAME AND ADDRESS Northrop Corporation Aircraft Division Hawthorne, CA 90250		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Washington, D.C.		12. ~~REPORT DATE~~ (11) October 1976
		13. NUMBER OF PAGES 378 (12) 381p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Air Development Center Warminster, Pennsylvania 18974		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approval for Public Release, Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

405 228

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Wave Drag, Three-Dimensional Flow, Supersonic Flow, Characteristics Methods, Aircraft Fuselage, Wing-Body Configuration, Optimization, Latin Squares, Numerical Method

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An optimization procedure has been developed to minimize the wave drag of an aircraft fuselage-wing configuration subject to constraints imposed by design requirements. The theory, methods, computer programs and results are presented in this report in two volumes. Volume I describes analyses, results and the optimization procedure. The procedure makes use of the Latin Square sampling technique and the Three-Dimensional Method of Characteristics. The former is used to efficiently sample the family of configurations, and the latter is used to accurately calculate the

next page

Jpg

20.

wave drags of the sampled configurations. The calculated wave drag coefficients are then used to derive a functional dependence of the wave drag on the geometric variables that define the family of configurations. The minimum wave drag configuration can be obtained by minimizing the wave drag function subject to a given set of constraints. The wave drag reduction procedure is demonstrated using an F-4 type configuration as the baseline. The results are presented and discussed.

This Volume ~~II~~ is the user's manual for the computer programs. The input/output information is described in detail. Listings of the programs are given, and samples of built-in program diagnostic messages are explained. Also included are the logical structures of the programs and the descriptions of the subroutines, which in combination with the program listings can be used for possible future modification, improvement, or extension of these computer programs.

FOREWORD

This work was performed by the Aerodynamics Research Organization of the Aircraft Division of the Northrop Corporation in Hawthorne, California for the Naval Air Development Center in Warminster, Pennsylvania (Contract No. N62269-75-C-0537) under the auspices of the Naval Air Systems Command. The contract monitor was Mr. Carmen Mazza of NADC, whose cooperation and assistance are gratefully acknowledged. The authors are indebted to Messrs. William Becker, David Bailey and K. T. Yen of NADC for their valuable assistance during the course of this work. Special thanks go to Messrs. Ray Siewert and Louis Schmidt of NASC for their encouragement and interest in the present work.

Preceding Page Blank - NOT FILMED

PREFACE

The wave drag reduction procedure presented in Volume I of this report makes use of three major computer programs: the Initial Value Surface (IVS) program, the Three-Dimensional Method of Characteristics (3DMoC) program, and the Surface Fit and Minimum Search program, which are described in the present volume in three parts. The procedure can be used to improve an existing aircraft or to aid in the design of a new one. In the case of improvement, the existing aircraft naturally serves as the baseline. In the case of new design, the preliminary configuration, which is usually obtained through considerations other than the wave drag, can serve as the baseline for wave drag reduction.

The wave drag reduction procedure consists of three major steps:

1. The first step is to select either the 5 x 5 Latin Square for the forward fuselage or the 3 x 3 Latin Square for the blended wing configuration and to describe the baseline and variation configurations using the body description method presented in Appendix C of Volume I. The sections on the description and variation of configuration in Appendix A or the main text of Volume I should be consulted in producing the body description. This step is time-consuming but must be done carefully to assure success in wave drag calculations by the method of characteristics.

2. For the next step of calculating the wave drag coefficients of the configurations sampled by each cell of the Latin Square, the IVS and 3DMoC programs are used. These two programs can run together with the IVS program providing the initial data needed by the 3DMoC program. Input cards can be prepared according to the instructions given in Parts 1 and 2 of the present volume and fed to the IVS and the 3DMoC programs for wave drag calculations. Care needs to be taken in the preparation of these cards, for if the calculation fails to proceed further the first item to check is the correctness of the input cards.

3. After the wave drag coefficients have been calculated by the 3DMoC program, the Surface Fit and Minimum Search Program can be used to define the minimum wave drag configuration for a given set of constraints as explained in Part 3 of the present volume. The program can be used to calculate various minimum wave drag configurations for different sets of constraints as long

as the same baseline configuration is considered and the variables stay within the ranges.

In the following parts, detailed information for setting up and using the three computer program are presented. Samples of deck set-up are compiled in the Appendix for illustration and reference.

TABLE OF CONTENTS

PART 1: THE INITIAL VALUE SURFACE PROGRAM

PART 2: THE THREE-DIMENSIONAL METHOD OF CHARACTERISTICS PROGRAM

ILLUSTRATIONS

Preceding Page Blank - Not Filmed

PART 1: THE INITIAL VALUE SURFACE PROGRAM

SECTION I

INTRODUCTION

The Initial Value Surface Program is used to generate starting data for the Three-Dimensional Method of Characteristics Program. In combination, these two programs determine the supersonic flow past smooth blunted bodies for an angle of attack at which the initial value surface remains on the blunted nose. Since the sonic point on a sphere lies between 40 and 50 degrees from the stagnation point, the sphere-body junction should occur at an angle greater than approximately $(50 - \alpha)$ degrees on the sphere, measured in the body axis system.

The lowest usable Mach number is around 2.5, with lower values attainable by careful attention to the initial value surface solution. There is no computational upper limit on the Mach number. However, above $M_\infty = 7.0$ (approximately) the ideal gas model becomes somewhat inaccurate.

The Initial Value Surface Program consists of four separate programs, all connected through an OVERLAY sequence. These programs are:

(1) Blunt Body Program
(2) Initial Value Line Interpolation Program
(3) Rotationally Symmetric Method of Characteristics Program
(4) Initial Value Surface Interpolation Program

The Blunt Body Program determines the subsonic through slightly supersonic flow over the blunted nose. The Initial Value Line Interpolation Program interpolates in the Blunt Body Program data to develop a start line for the Rotationally Symmetric Method of Characteristics Program. This latter program calculates the required remaining supersonic flow field over the nose. The Initial Value Surface Interpolation Program interpolates in the RSMoC-generated data to determine properties on a plane normal to the body axis, given the number of data rings and the number of meridian planes.

The final output is provided in printed form and is written on a scratch tape for use

by the Three-Dimensional Method of Characteristics Program.

The Initial Value Surface Program consists of approximately 3000 cards. It is written entirely in FORTRAN EXTENDED and has been run under OPT = 2 on the CDC FTN Compiler.

SECTION II

INPUT/OUTPUT DESCRIPTION

1. INPUT DESCRIPTION*

The input data cards required by the Initial Value Surface Program are shown in Figure 1. They are described in detail below. The limits indicated refer to storage limitations of the program in its present form.

Card No.	Variable	Format	Description
①	TITLE	7A10	Problem identification
②	ACHM	E12.4	Freestream Mach number
	GAMA	E12.4	Ratio of specific heats
	ALFA	E12.4	Angle of attack, in degrees
	ELAM	E12.4	Slope of the sphere at the sphere-body juncture, in degrees. For bodies with non-conical shapes, this slope should be the largest value of the slopes around the sphere-body juncture.
	RINGS	E12.4	Number of data rings on the initial value surface. A minimum of 3 is recommended. Limit: RINGS < 20
	ONE05	E12.4	M_{body}, Mach number on the body at the point through which the IVL is passed. For $M_{\infty} \geq 2.5$, a value of 1.05 is recommended for M_{body}. For lower Mach numbers, M_{body} should be increased to 1.10. The upper limit is the highest Mach number calculated by the Blunt Body portion of the program on the surface of the sphere. When M_{body} is specified to be greater than this highest value, the program automatically reduces M_{body} to .01 below this value.

*The Appendix may be consulted for sample deck setup.

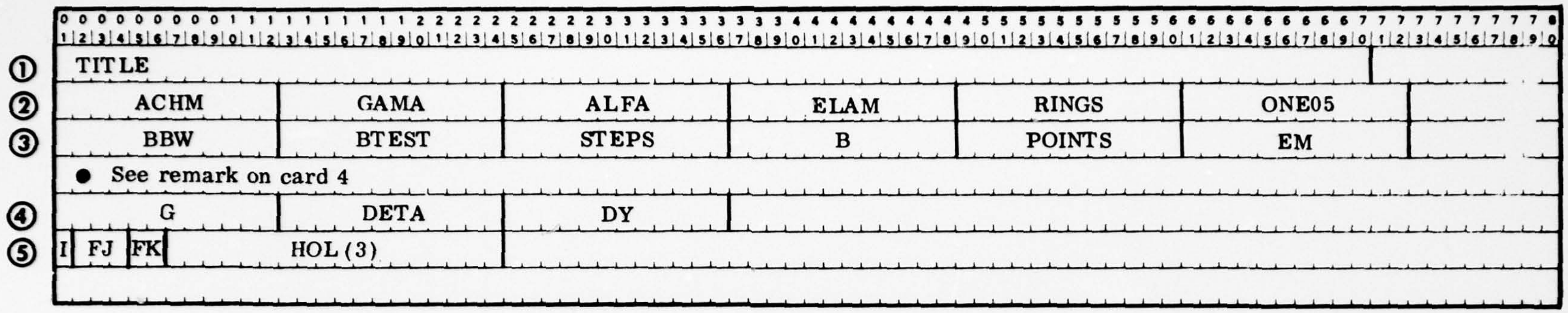


FIGURE 1. INPUT DATA FOR INITIAL VALUE SURFACE PROGRAM

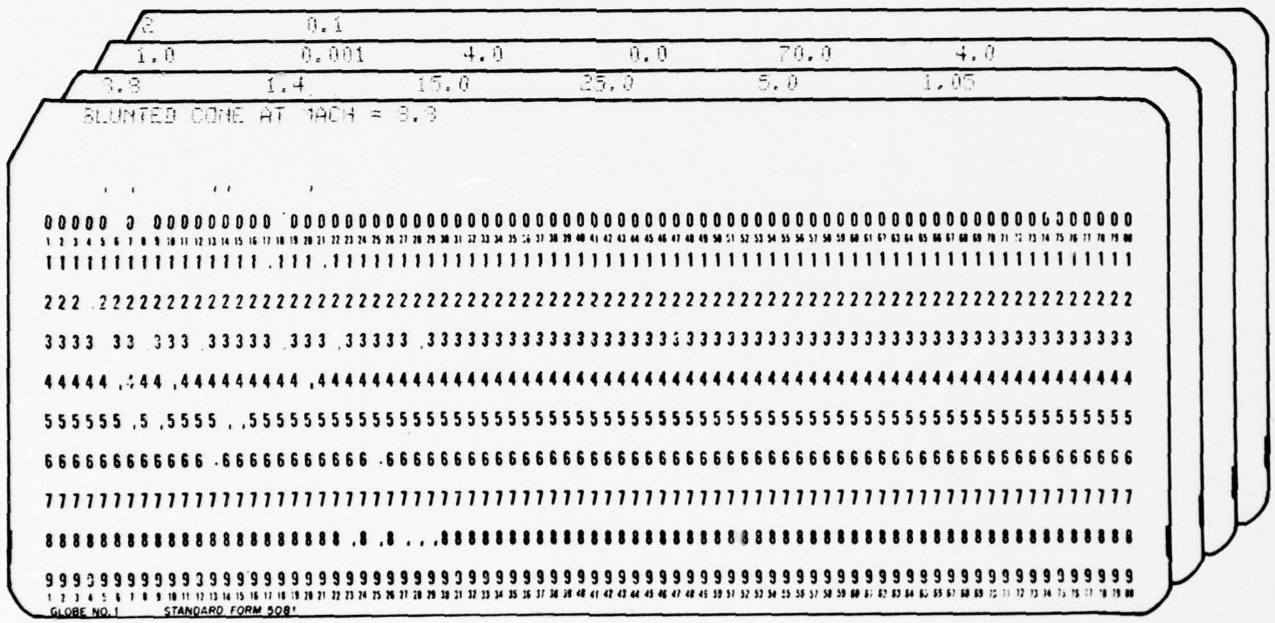


FIGURE 2. SAMPLE INPUT FOR 25° CONE AT 15° ANGLE OF ATTACK

Card No.	Variable	Format	Description
3	BBW	E12.4	Body bluntness, B_{body}. For a sphere, B_{body} is equal to 1.0. For an ellipse, B_{body} is equal to a^2/b^2, where a and b are the semi-axes of the ellipse.
	BTEST	E12.4	The tolerance, ΔB, accepted in B_{body}. A value of .001 is usually sufficient.
	STEPS	E12.4	Number of steps required to reach the body, not counting the shock wave. For $M_\infty \geqq 2.5$, a value of 4 should be sufficient. For lower Mach numbers, larger values should be used. Limit: STEPS < 10
	B	E12.4	Initial guess at the shock wave bluntness, B_{shock}. If B = 0, the program computes its own value based on an approximating curve. Card 4 is not required for this case.
	POINTS	E12.4	Number of shock points. For $M_\infty > 2.5$, a value of 70 should be sufficient. For lower Mach numbers, this value should be increased. Limit: POINTS < 100
	EM	E12.4	Number of meridian planes in 90°. This defines the spacing of the meridian planes. Note that, if EM is the number of meridian planes in 90°, 2EM-1 is the number in 180°. Limit: EM ≦ 10

- If B = 0, omit card 4

Card No.	Variable	Format	Description
4	G	E12.4	Step size multiplier, to be obtained from previous calculations or other knowledge.
	DETA	E12.4	Initial step size normal to the shock, $\Delta\eta$
	DY	E12.4	Initial step size along the shock, $\Delta\xi$
5	I	I1	Output control for the RSMoC portion of the program. I = 2 is recommended. If I = 1, only body points are printed; = 2, all points are printed; = 4, detailed print is generated
	FJ	F3.0	Initial C-Summary Number for obtaining local checkout print. Should normally be zero.

Card No.	Variable	Format	Description
5	FK	F2.0	Final C-Summary Number for obtaining local checkout print. Should normally be zero.
	HOL (3)	E18.6	Step size along the body for the RSMoC portion of the program. A value of 0.1 is recommended. Should difficulties arise in the RSMoC portion of the program, a small change in this value will rearrange the mesh points and often cure the problem.

Figure 2 shows the input data cards required to generate an initial value surface for a blunted 25^o cone at 15^o angle of attack.

2. OUTPUT DESCRIPTION

The first item of output is a listing of the input data, as read. Figure 3 is an example of such output. Whenever unusual messages or stops occur later in the program, this listing should be examined for correctness. Also note that an interpolation Mach number which is higher than the recommended value (for the freestream Mach number specified) has been used. This was done to exercise the automatic resetting feature of the program, which was discussed in the preceding section.

Each iteration of the Blunt Body portion of the program generates output, such as shown in Figure 4. At the top are given the freestream conditions, followed by the governing quantities for the Blunt Body portion of the program, B, DETA, DY and G which have all been identified in the preceding section. Following this is a progress report showing the number of points on each shock-like line. Note that the program will drop a minimum of six points in passing from line to line. More points may be dropped due to increasing roughness in the interpolation process. The results of the body fit process are listed, where DELTA is the shock stand-off distance, Δ/R_{shock}, B(BODY) is the body bluntness, and R(BODY) is the ratio of R_{body}/R_{shock}. The η and ξ coordinates and the local Mach number are given for each subsonic body point. Near convergence, the final body Mach number should be only slightly less than 1.0.

When convergence has occurred, the complete blunt body flow field is printed out, with the coordinates now normalized by R_{body}. Figure 5 shows a typical first page of such an output. At the top are given the freestream conditions, and the geometric

```
* * * * * * * * * * * * * *  LEWIS CHECK CASE NO.1F  %TIDY DECK CHECK<  MAY/10/72      * * * * * * * * * * * * * * * * *

                       *****
                       * I *  MACH NO.                   =  8.8000
                       * N *  GAMMA                      =  1.4000
                       * P *  ANGLE OF ATTACK            =  15.000     DEG.
                       * U *  BODY SLOPE                 =  25.000     DEG.
                       * T *  NO. OF RINGS               =  5.0000
                       *   *  INTERPOLATION MACH NO.     =  1.1500
                       * D *  BODY BLUNTNESS             =  1.0000
                       * A *  BLUNTNESS TOLERANCE        =  .10000E-02
                       * T *  STEPS                      =  4.0000
                       * A *  B(SHOCK)                   =  -0.
                       * : *  NO. OF POINTS              =  70.000
                       *****  NO. OF MERIDIANS           =  4.0000
```

FIGURE 3. LISTING OF INPUT DATA

```
                            FREE STREAM CONDITIONS
               M=      .880000E+01
               GAMMA=  1.400000
    PARAMETERS              H=          .451491E+00
                            DETA=       .207438E-01
                            DY=         .200000E-01
                            G=          .100000E+01
    ETA STEP NO    1        ETA= 1.0000000      NO OF XI S = 70
    ETA STEP NO    2        ETA=  .9792562      NO OF XI S = 62
    ETA STEP NO    3        ETA=  .9585124      NO OF XI S = 55
    ETA STEP NO    4        ETA=  .9377686      NO OF XI S = 49
    ETA STEP NO    5        ETA=  .9170248      NO OF XI S = 43
    ETA STEP NO    6        ETA=  .8962810      NO OF XI S = 37
****************************************************************************************************
****************************************************************************************************

    DELTA=     .1031314E+00
  B(BODY)=     .8519421E+00
  R(BODY)=     .7433888E+00
****************************************************************************************************
****************************************************************************************************
          NO.                ETA                 XI                  MACH
           1            .893619E+00         .100000E-01         .203352E-01
           2            .893566E+00         .300000E-01         .483754E-01
           3            .893461E+00         .500000E-01         .787246E-01
           4            .893302E+00         .700000E-01         .109517E+00
           5            .893091E+00         .900000E-01         .140506E+00
           6            .892825E+00         .110000E+00         .171634E+00
           7            .892504E+00         .130000E+00         .202892E+00
           8            .892128E+00         .150000E+00         .234284E+00
           9            .891694E+00         .170000E+00         .265825E+00
          10            .891203E+00         .190000E+00         .297533E+00
          11            .890653E+00         .210000E+00         .329429E+00
          12            .890041E+00         .230000E+00         .361538E+00
          13            .889365E+00         .250000E+00         .393887E+00
          14            .888625E+00         .270000E+00         .426507E+00
          15            .887816E+00         .290000E+00         .459435E+00
          16            .886936E+00         .310000E+00         .492709E+00
          17            .885982E+00         .330000E+00         .526374E+00
          18            .884950E+00         .350000E+00         .560483E+00
          19            .883835E+00         .370000E+00         .595096E+00
          20            .882632E+00         .390000E+00         .630282E+00
          21            .881337E+00         .410000E+00         .666126E+00
          22            .879942E+00         .430000E+00         .702726E+00
          23            .878440E+00         .450000E+00         .740200E+00
          24            .876821E+00         .470000E+00         .778697E+00
```

FIGURE 4. INTERMEDIATE BLUNT BODY OUTPUT

COMPLETE FLOW FIELD NORMALIZED BY RBODY

M= 8.800000
GAMA= 1.400000
RB= .100000E+01
R= .100000E+01
DELTA= .137283E+00

X	R	MACH NO	VELOCITY	TOT PRES	PRESSURE	TEMPERATURE	DENSITY	DELTA	THETA
.633329E-04	.129824E-01	.3909266	.177688E+00	.550484E-02	.901714E+02	.159990E+02	.563607E+01	2.58852	89.44099
.570052E-03	.389473E-01	.3956237	.179759E+00	.551445E-02	.901026E+02	.159875E+02	.563581E+01	7.69219	88.32308
.158379E-02	.649122E-01	.4048641	.183826E+00	.553374E-02	.899652E+02	.159646E+02	.563528E+01	12.58567	87.20545
.310514E-02	.908770E-01	.4183654	.189752E+00	.556279E-02	.897595E+02	.159303E+02	.563450E+01	17.15888	86.08830
.513501E-02	.116842E+00	.4357583	.197357E+00	.560176E-02	.894860E+02	.158847E+02	.563345E+01	21.33741	84.97181
.767460E-02	.142807E+00	.4566327	.206441E+00	.565084E-02	.891451E+02	.158279E+02	.563214E+01	25.08290	83.85618
.107254E-01	.168772E+00	.4805762	.216799E+00	.571030E-02	.887378E+02	.157600E+02	.563056E+01	28.38731	82.74159
.142893E-01	.194737E+00	.5072014	.228238E+00	.578044E-02	.882648E+02	.156812E+02	.562871E+01	31.26463	81.62821
.183684E-01	.220701E+00	.5361614	.240579E+00	.586164E-02	.877272E+02	.155916E+02	.562658E+01	33.74258	80.51622
.229651E-01	.246666E+00	.5671563	.253668E+00	.595432E-02	.871261E+02	.154914E+02	.562417E+01	35.85596	79.40578
.280823E-01	.272631E+00	.5999329	.267368E+00	.605899E-02	.864629E+02	.153808E+02	.562147E+01	37.64195	78.29706
.337232E-01	.298596E+00	.6342815	.281565E+00	.617623E-02	.857388E+02	.152601E+02	.561848E+01	39.13708	77.19020
.398911E-01	.324561E+00	.6700308	.296159E+00	.630667E-02	.849554E+02	.151295E+02	.561520E+01	40.37554	76.08535
.465899E-01	.350526E+00	.7070426	.311067E+00	.645106E-02	.841142E+02	.149893E+02	.561160E+01	41.38836	74.98265
.538239E-01	.376491E+00	.7452071	.326218E+00	.661021E-02	.832171E+02	.148398E+02	.560770E+01	42.20319	73.88222
.615977E-01	.402455E+00	.7844378	.341552E+00	.678505E-02	.822657E+02	.146812E+02	.560347E+01	42.84435	72.78418
.699163E-01	.428420E+00	.8246683	.357017E+00	.697660E-02	.812619E+02	.145139E+02	.559891E+01	43.33307	71.68864
.787851E-01	.454385E+00	.8658486	.372568E+00	.718604E-02	.802077E+02	.143382E+02	.559400E+01	43.68777	70.59568
.882101E-01	.480350E+00	.9079428	.388169E+00	.741463E-02	.791050E+02	.141543E+02	.558874E+01	43.92447	69.50540
.981975E-01	.506315E+00	.9509269	.403785E+00	.766383E-02	.779559E+02	.139628E+02	.558311E+01	44.05702	68.41786
.108754E+00	.532280E+00	.9947873	.419390E+00	.793521E-02	.767624E+02	.137639E+02	.557710E+01	44.09742	67.33312
.119887E+00	.558245E+00	1.0395192	.434956E+00	.823058E-02	.755267E+02	.135579E+02	.557069E+01	44.05610	66.25122
.131605E+00	.584210E+00	1.0851254	.450464E+00	.855191E-02	.742510E+02	.133452E+02	.556387E+01	43.94212	65.17219
.143916E+00	.610174E+00	1.1316162	.465893E+00	.890144E-02	.729373E+02	.131262E+02	.555661E+01	43.76336	64.09605
.156828E+00	.636139E+00	1.1790079	.481228E+00	.928163E-02	.715879E+02	.129013E+02	.554890E+01	43.52670	63.02279
.170353E+00	.662104E+00	1.2273231	.496452E+00	.969526E-02	.702050E+02	.126708E+02	.554071E+01	43.23816	61.95239
.184499E+00	.688069E+00	1.2765899	.511554E+00	.101454E-01	.687907E+02	.124350E+02	.553203E+01	42.90299	60.88483
.199278E+00	.714034E+00	1.3268419	.526521E+00	.106357E-01	.673472E+02	.121943E+02	.552282E+01	42.52582	59.82005
.214703E+00	.739999E+00	1.3781182	.541344E+00	.111698E-01	.658766E+02	.119492E+02	.551305E+01	42.11071	58.75798
.230784E+00	.765964E+00	1.4304632	.556013E+00	.117523E-01	.643810E+02	.116999E+02	.550271E+01	41.66123	57.69854
.247538E+00	.791929E+00	1.4839271	.570520E+00	.123881E-01	.628626E+02	.114468E+02	.549174E+01	41.18053	56.64162
.264977E+00	.817893E+00	1.5385655	.584859E+00	.130827E-01	.613235E+02	.111902E+02	.548012E+01	40.67137	55.58709
.283117E+00	.843858E+00	1.5944401	.599024E+00	.138426E-01	.597655E+02	.109304E+02	.546781E+01	40.13621	54.53480
.301976E+00	.869823E+00	1.6516192	.613009E+00	.146749E-01	.581908E+02	.106679E+02	.545475E+01	39.57718	53.48458
.321571E+00	.895788E+00	1.7101777	.626810E+00	.155878E-01	.566012E+02	.104029E+02	.544091E+01	38.99618	52.43624
.341922E+00	.921753E+00	1.7701980	.640423E+00	.165905E-01	.549987E+02	.101357E+02	.542622E+01	38.39487	51.38955
.363049E+00	.947718E+00	1.8317705	.653844E+00	.176937E-01	.533850E+02	.986669E+01	.541063E+01	37.77470	50.34428
.384975E+00	.973683E+00	1.8949945	.667071E+00	.189095E-01	.517619E+02	.959608E+01	.539406E+01	37.13693	49.30014
.407724E+00	.999648E+00	1.9599790	.680102E+00	.202520E-01	.501311E+02	.932419E+01	.537446E+01	36.48264	48.25684
.431322E+00	.102561E+01	2.0266439	.692936E+00	.217373E-01	.484944E+02	.905129E+01	.535773E+01	35.81275	47.21403
.455797E+00	.105158E+01	2.0957214	.705569E+00	.233839E-01	.468532E+02	.877764E+01	.533779E+01	35.12805	46.17134
.481181E+00	.107754E+01	2.1667570	.718003E+00	.252136E-01	.452091E+02	.850350E+01	.531653E+01	34.42917	45.12836
.507507E+00	.110351E+01	2.2401117	.730236E+00	.272514E-01	.435635E+02	.822910E+01	.529384E+01	33.71661	44.08464
.534810E+00	.112947E+01	2.3159640	.742268E+00	.295269E-01	.419179E+02	.795470E+01	.526958E+01	32.99075	43.03967
.563131E+00	.115544E+01	2.3945121	.754100E+00	.320745E-01	.402737E+02	.768049E+01	.524363E+01	32.25183	41.99291
.592513E+00	.118140E+01	2.4759769	.765731E+00	.349349E-01	.386313E+02	.740670E+01	.521581E+01	31.49999	40.94375

FIGURE 5. FINAL BLUNT BODY OUTPUT

properties of the flow field, R_{body}, B_{body}, and Δ/R_{body}.

Data along each shock-like coordinate line are printed out, starting with the shock itself. The solution is located in a Cartesian coordinate system with the origin located at the stagnation point on the shock (i.e., the intersection of the bow shock and the stagnation streamline). X and R are the abscissa and ordinate of the point, respectively, MACH NO is the Mach number, VELOCITY is V/V_∞ , DELTA is the local flow direction in degrees, and THETA is the local inclination of the shock wave in degrees.

Figure 6 shows a typical final page output which represents the body. Note that the total pressure values are all identical. This is the distinguishing feature of the body point output. Also note that the maximum body surface Mach number is 1.0910140.

X	R	MACH NO	VELOCITY	TOT PRES	PRESSURE	TEMPERATURE	DENSITY	DELTA
.135813E+00	.115518E-01	.0146529	.672825E-02	.368284E-02	.119105E+03	.194312E+02	.612957E+01	88.61040
.136350E+00	.346535E-01	.0450707	.206916E-01	.368284E-02	.118953E+03	.194241E+02	.612401E+01	87.75170
.137425E+00	.577495E-01	.0752941	.345543E-01	.368284E-02	.118651E+03	.194100E+02	.611289E+01	86.50845
.139037E+00	.808361E-01	.1055335	.484055E-01	.368284E-02	.118199E+03	.193888E+02	.609622E+01	85.20960
.141190E+00	.103909E+00	.1358292	.622560E-01	.368284E-02	.117597E+03	.193606E+02	.607404E+01	83.89142
.143883E+00	.126965E+00	.1662070	.761098E-01	.368284E-02	.116847E+03	.193252E+02	.604637E+01	82.56355
.147120E+00	.150000E+00	.1966900	.899697E-01	.368284E-02	.115952E+03	.192828E+02	.601324E+01	81.22946
.150902E+00	.173009E+00	.2273011	.103838E+00	.368284E-02	.114913E+03	.192333E+02	.597469E+01	79.89053
.155233E+00	.195988E+00	.2580643	.117718E+00	.368284E-02	.113732E+03	.191766E+02	.593077E+01	78.54725
.160115E+00	.218933E+00	.2890047	.131612E+00	.368284E-02	.112412E+03	.191127E+02	.588152E+01	77.19965
.165554E+00	.241839E+00	.3201493	.145524E+00	.368284E-02	.110956E+03	.190417E+02	.582700E+01	75.84748
.171553E+00	.264701E+00	.3515276	.159458E+00	.368284E-02	.109367E+03	.189633E+02	.576726E+01	74.49028
.178117E+00	.287514E+00	.3831713	.173419E+00	.368284E-02	.107647E+03	.188777E+02	.570235E+01	73.12743
.185253E+00	.310272E+00	.4151159	.187413E+00	.368284E-02	.105801E+03	.187846E+02	.563233E+01	71.75813
.192965E+00	.332969E+00	.4474006	.201447E+00	.368284E-02	.103931E+03	.186840E+02	.555723E+01	70.38142
.201260E+00	.355600E+00	.4800695	.215529E+00	.368284E-02	.101741E+03	.185758E+02	.547710E+01	68.99614
.210148E+00	.378157E+00	.5131724	.229670E+00	.368284E-02	.995342E+02	.184597E+02	.539196E+01	67.60093
.219635E+00	.400632E+00	.5467665	.243882E+00	.368284E-02	.972129E+02	.183357E+02	.530184E+01	66.19414
.229731E+00	.423018E+00	.5804176	.258178E+00	.368284E-02	.947799E+02	.182034E+02	.520672E+01	64.77386
.240446E+00	.445305E+00	.6157022	.272577E+00	.368284E-02	.922377E+02	.180625E+02	.510658E+01	63.33779
.251791E+00	.467484E+00	.6512110	.287099E+00	.368284E-02	.895878E+02	.179127E+02	.500135E+01	61.88319
.263780E+00	.489543E+00	.6875518	.301770E+00	.368284E-02	.868312E+02	.177535E+02	.489094E+01	60.40679
.276426E+00	.511470E+00	.7246550	.316622E+00	.368284E-02	.839677E+02	.175842E+02	.477518E+01	58.90460
.289744E+00	.533251E+00	.7632804	.331695E+00	.368284E-02	.809957E+02	.174041E+02	.465383E+01	57.37182
.303753E+00	.554869E+00	.8030274	.347038E+00	.368284E-02	.779121E+02	.172121E+02	.452658E+01	55.80248
.318470E+00	.576307E+00	.8443486	.362715E+00	.368284E-02	.747110E+02	.170071E+02	.439294E+01	54.18922
.333920E+00	.597543E+00	.8875710	.378808E+00	.368284E-02	.713833E+02	.167871E+02	.425227E+01	52.52272
.350127E+00	.618551E+00	.9331278	.395428E+00	.368284E-02	.679153E+02	.165499E+02	.410366E+01	50.79112
.367121E+00	.639303E+00	.9816078	.412722E+00	.368284E-02	.642867E+02	.162923E+02	.394583E+01	48.97898
.384936E+00	.659763E+00	1.0338365	.430896E+00	.368284E-02	.604679E+02	.160097E+02	.377695E+01	47.06582
.403611E+00	.679888E+00	1.0910140	.450243E+00	.368284E-02	.564151E+02	.156955E+02	.359435E+01	45.02390

FIGURE 6. FLOW PROPERTIES ON THE BODY

In Figure 3, the input data listed an interpolation Mach number of 1.15, which is greater than the last body surface Mach number.

Figure 7 shows the sonic line and stagnation streamline data derived by interpolation. Along the stagnation line, the local flow angle should be zero, except at the last point (the body stagnation point), where a singularity exists. The results shown in Figure 7 represent reasonable values.

Figure 8 shows the resulting initial value line obtained by interpolation in the blunt body data. The initial message is caused by the fact that the interpolation Mach number was specified to be larger than the maximum Mach number on the body surface, calculated in the Blunt Body portion of the program. When this occurs, the program automatically resets the interpolation Mach number to be less than the final body surface Mach number.

The listing of the freestream properties and geometric properties of the blunt body solution is followed by the interpolated drag coefficient, the table in which the interpolation was carried out, and the value used for interpolation.

SONIC POINT DATA

X	R	MACH NO	VELOCITY	TOT PRES	PRESSURE	TEMPERATURE	DENSITY	DELTA	THETA
.110033E+00	.535332E+00	1.0000010	.421222E+00	.796869E-02	.766193E+02	.137400E+02	.699786E+03	44.09680	67.20578
.155691E+00	.570806E+00	1.0000009	.421222E+00	.777957E-02	.748010E+02	.137400E+02	.699787E+03	45.80217	
.208209E+00	.604705E+00	1.0000018	.421222E+00	.740177E-02	.711685E+02	.137400E+02	.699787E+03	46.91614	
.266752E+00	.633135E+00	1.0000043	.421223E+00	.683925E-02	.657598E+02	.137400E+02	.699786E+03	47.51264	
.327921E+00	.648910E+00	1.0000117	.421226E+00	.613318E-02	.589705E+02	.137400E+02	.699784E+03	47.92760	
.375448E+00	.647354E+00	1.0000373	.421236E+00	.550363E-02	.529170E+02	.137399E+02	.699779E+03	48.24784	

STAGNATION POINT DATA

X	R	MACH NO	VELOCITY	TOT PRES	PRESSURE	TEMPERATURE	DENSITY	DELTA	THETA
.105068E-06	.111022E-15	.3902817	.173444E+00	.550365E-02	.901801E+02	.152942E+02	.102407E+04	-.04213	90.00000
.262717E-01	0.	.3068528	.137435E+00	.550365E-02	.938346E+02	.155346E+02	.105354E+04	-.07108	
.568840E-01	0.	.2170013	.980253E-01	.550364E-02	.969318E+02	.158022E+02	.107827E+04	-.14715	
.924622E-01	.111022E-15	.1184626	.540254E-01	.550364E-02	.991820E+02	.161064E+02	.109609E+04	-.44152	
.133673E+00	-.111022E-15	.0013815	.637361E-03	.550364E-02	.100168E+03	.164834E+02	.110386E+04	42.66628	
.137210E+00	-.121426E-05	.0009386	.433041E-03	.550363E-02	.100169E+03	.164849E+02	.110387E+04	88.90918	

FIGURE 7. SONIC LINE AND STAGNATION STREAMLINE

*INTERPOLATION MACH NO. DECREASED TO 1.090 DUE TO LIMITATION ON BODY MACH NO.

INITIAL VALUE LINE DATA

MACH NO.	GAMMA
.8800000E+01	.1400000E+01

DELTA	RBODY	RBODY
.1372834E+00	.1000000E+01	.1000000E+01

CD= .622474E+00 BASED ON THE FOLLOWING TABLE

Y(I)	CPR(I)
.682594E+00	.625498E+00
.662436E+00	.601610E+00
.641935E+00	.576398E+00

INTERPOLATED FOR Y= .679999E+00

IVL DATA.STARTING AT SHOCK

X	Y	DELTA	MACH NO.	PT/PTO
.2623376E+00	.8140287E+00	.4074896E+02	.1530358E+01	.1297562E-01
.2835064E+00	.7943341E+00	.4143026E+02	.1421652E+01	.1083439E-01
.3082310E+00	.7713309E+00	.4230574E+02	.1318021E+01	.9064620E-02
.3370515E+00	.7445169E+00	.4348350E+02	.1222897E+01	.7610691E-02
.3705602E+00	.7133409E+00	.4516656E+02	.1143817E+01	.6422311E-02
.4051535E+00	.6799995E+00	.4711418E+02	.1089626E+01	.5503634E-02

MASS-ENTROPY TABLE

0.	.5503634E-02	.1298244E-01	.5504835E-02	.3894731E-01	.5514453E-02
.6491218E-01	.5533738E-02	.9087705E-01	.5562789E-02	.1168419E+00	.5601756E-02
.1428068E+00	.5650840E-02	.1687717E+00	.5710297E-02	.1947365E+00	.5780439E-02
.2207014E+00	.5861636E-02	.2466663E+00	.5954321E-02	.2726311E+00	.6058995E-02
.2985960E+00	.6176229E-02	.3245609E+00	.6306672E-02	.3505258E+00	.6451057E-02
.3764906E+00	.6610207E-02	.4024555E+00	.6785045E-02	.4284204E+00	.6976605E-02
.4543852E+00	.7186039E-02	.4803501E+00	.7414634E-02	.5063150E+00	.7663826E-02
.5322798E+00	.7935213E-02	.5582447E+00	.8230579E-02	.5842096E+00	.8551913E-02
.6101745E+00	.8901436E-02	.6361393E+00	.9281628E-02	.6621042E+00	.9695262E-02
.6880691E+00	.1014544E-01	.7140339E+00	.1063566E-01	.7399988E+00	.1116981E-01
.7659637E+00	.1175230E-01	.7919285E+00	.1238808E-01	.8140287E+00	.1297562E-01

FIGURE 8. INITIAL VALUE LINE

The final table is the mass-entropy table which is a listing of the shock point ordinates (which define the swallowed mass) and the associated total pressures. Each line consists of three pairs of data, with each pair being a shock ordinate and total pressure. The first pair represents the stagnation streamline and the final pair the initial value line shock point.

Figure 9 shows the first page of the Rotationally Symmetric Method of Characteristics output. The body geometric description occupies the upper portion of the output. The body is now located in a coordinate system centered at the body stagnation point. The blunt segment equation is then

$$Y = \sqrt{2RX - BX}$$

where the values of R and B are given in the output. The second section is a specialization of the general cubic equation

$$Y = A(X-X_1)^3 + B(X-X_1)^2 + C(X-X_1) + D$$

```
                                                                                                        PAGE    1
 BLUNT SEGMENT  1  X2=  .826352E+00    Y2=  .984808E+00    X*=  .100164E+00
                           R=  .100000E+01   B=  .100000E+01
 CUBIC SEGMENT  2  X2=  .200000E+01    Y2=  .119175E+01    X*=  .104324E+00
                        SLOPE 1=  10.00000 DEG   SLOPE 2=  10.00000 DEG
                        A= 0.               B= 0.              C=  .176327E+00     D=  .984808E+00
*******************************************ROTATIONALLY SYMMETRIC FLOW*******************************************

                                             33 ENTROPY VALUES READ

----C-SUMMARY NO.  1-------------------------------------------------------------------------------------------

POINT        X            Y           DELTA      MACH        P/PTI        PT/PTO

   1     .233277      .713341       45.167     1.1438      .937454      .642231E-02
   2     .258242      .706202       45.282     1.1389      .873937      .595844E-02
   3     .283815      .697910       45.740     1.0985      .848606      .550363E-02

            DRAG COEF.=  .643534E+00                       (DELTA M)/(M TOTAL)=    .422081E-01
---------------------------------------------------------------------------------------------------------------

----C-SUMMARY NO.  2-------------------------------------------------------------------------------------------

POINT        X            Y           DELTA      MACH        P/PTI        PT/PTO

   1     .199768      .744517       43.483     1.2229      1.00415      .761069E-02
   2     .227292      .738517       43.699     1.1935      .950355      .693776E-02
   3     .251657      .732613       43.646     1.1835      .890420      .642490E-02
   4     .274805      .726505       44.072     1.1592      .856779      .599684E-02
   5     .303571      .717626       44.141     1.1233      .822257      .550363E-02

            DRAG COEF.=  .666844E+00                       (DELTA M)/(M TOTAL)=    .111710E-01
---------------------------------------------------------------------------------------------------------------

----C-SUMMARY NO.  3-------------------------------------------------------------------------------------------

POINT        X            Y           DELTA      MACH        P/PTI        PT/PTO

   1     .170948      .771331       42.306     1.3180      1.05331      .906462E-02
   2     .197288      .767619       42.458     1.2785      1.00756      .822413E-02
   3     .223760      .763010       42.521     1.2477      .956394      .749726E-02
   4     .247648      .758249       42.338     1.2341      .899161      .693122E-02
   5     .269548      .753393       42.609     1.2087      .867818      .647396E-02
   6     .296566      .746740       42.780     1.1893      .820562      .597573E-02
   7     .325773      .738524       42.394     1.1622      .781742      .550363E-02

            DRAG COEF.=  .691241E+00                       (DELTA M)/(M TOTAL)=    .781880E-03
---------------------------------------------------------------------------------------------------------------
```

FIGURE 9. ROTATIONALLY SYMMETRIC METHOD OF CHARACTERISTICS OUTPUT

where A, B, C, D are given in the output. For each segment, X2 , Y2 are the coordinates of the final point. For segment 1, these coordinates represent the juncture between the sphere and the cone of slope $\lambda - \alpha$ (body slope minus angle of attack).

Next is printed a reminder that this is rotationally symmetric flow, followed by the number of mass-entropy points read. This should correspond to the number prepared by the Initial Value Line portion of the program.

The data for each characteristic line are grouped under C-SUMMARY headings which are numbered consecutively. The first few C-Summaries will start from points on the IVL and these will not have shock points. Eventually, however, the C-Summaries will have shock points as Point 1. This is denoted by the value of the shock angle being printed out as THETA SHOCK 1 = XXX just beneath the C-Summary number and by a 1 being printed at the extreme right of the shock point data.

Each C-Summary contains, as a final item, the summed wave drag coefficient and the mass flow error. After the first few C-Summaries, this mass flow error should be much less than 0.10. Otherwise, a smaller step size is needed.

```
        ANGLE OF ATTACK =15.00     CONE ANGLE =    25.00000     DATA PLANE =    .5773817

  8.800000  1.400000 15.000000 .57738174
RING NO.  1
     Z              X              MACH           PT/PTO        THETA          PSI
 -.9063078E+00  -.1726768E-11 1.2217258     .5503634E-02 24.99999979-90.00000000
 -.8181791E+00   .3936596E+00 1.2883293     .5503634E-02 26.01400608-45.34219559
 -.5462947E+00   .7293347E+00 1.4203431     .5503634E-02 27.96334537 -5.19490444
 -.1093817E+00   .9063078E+00 1.6273434     .5503634E-02 28.90455539 29.03199324
  .3836752E+00   .8265352E+00 1.8940439     .5503634E-02 27.96334537 56.54005111
  .7649038E+00   .4891902E+00 2.1225041     .5503634E-02 26.01400608 76.36295185
  .9063078E+00  -.2219890E-11 2.1932926     .5503634E-02 24.99999979 90.00000000
RING NO.  2
     Z              X              MACH           PT/PTO        THETA          PSI
 -.9723198E+00  -.1870498E-11 1.3527225     .6762385E-02 23.46334930-90.00000000
 -.8783550E+00   .4272183E+00 1.4051956     .6802801E-02 25.04536870-44.63441771
 -.5866148E+00   .7967915E+00 1.5603617     .7006888E-02 28.26905526 -5.54567256
 -.1093817E+00   .1003121E+01 1.7605251     .7230446E-02 31.37634749 26.06426598
  .4462869E+00   .9312867E+00 1.9743206     .7376107E-02 32.75191677 51.81681644
  .8923473E+00   .5602625E+00 2.1299215     .7456217E-02 33.00740706 72.44252829
  .1062272E+01  -.2559474E-11 2.1854259     .7485228E-02 32.97651630 90.00000000
RING NO.  3
     Z              X              MACH           PT/PTO        THETA          PSI
 -.1038332E+01  -.2014228E-11 1.5055606     .8831296E-02 22.28056799-90.00000000
 -.9385309E+00   .4607771E+00 1.5614025     .9083405E-02 24.17716680-43.95265369
 -.6269348E+00   .8642483E+00 1.7257297     .9782160E-02 28.09339206 -5.34510087
 -.1093817E+00   .1099934E+01 1.9429379     .1081324E-01 32.07705509 25.31077296
  .5088987E+00   .1036038E+01 2.1627564     .1182286E-01 34.62248828 50.32562417
  .1019791E+01   .6313349E+00 2.3247610     .1256621E-01 35.78567131 71.28336665
  .1218235E+01  -.2899059E-11 2.3828647     .1283465E-01 36.10582939 90.00000000
RING NO.  4
     Z              X              MACH           PT/PTO        THETA          PSI
 -.1104344E+01  -.2157957E-11 1.7138772     .1275552E-01 21.57700346-90.00000000
 -.9987068E+00   .4943358E+00 1.7769999     .1342631E-01 23.74117593-43.59165881
 -.6672549E+00   .9317050E+00 1.9601956     .1552433E-01 28.36124020 -5.64988508
 -.1093817E+00   .1196747E+01 2.2342665     .1926272E-01 32.77665704 24.59163242
  .5715104E+00   .1140790E+01 2.5009086     .2365976E-01 36.03404669 49.29875027
  .1147234E+01   .7024072E+00 2.6937877     .2728776E-01 37.76813957 70.54981016
  .1374199E+01  -.3238643E-11 2.7620758     .2851835E-01 38.30402320 90.00000000
RING NO.  5
     Z              X              MACH           PT/PTO        THETA          PSI
 -.1170356E+01  -.2301687E-11 1.9963238     .2104778E-01 21.11800828-90.00000000
 -.1058883E+01   .5278946E+00 2.0764296     .2290993E-01 23.40411353-43.30339389
 -.7075749E+00   .9991618E+00 2.2785695     .2638318E-01 28.59447790 -5.91041171
 -.1093817E+00   .1293559E+01 2.5971164     .3962772E-01 33.58825526 23.79562061
  .6341222E+00   .1245541E+01 2.9389790     .5610073E-01 37.17580737 48.52184246
  .1274678E+01   .7734796E+00 3.1780746     .7090419E-01 39.33140274 70.01710784
  .1530163E+01  -.3578227E-11 3.2680198     .7726594E-01 40.00518452 90.00000000
```

FIGURE 10. INITIAL VALUE SURFACE

When the drag coefficient and the mass flow error no longer change, the C-Summaries no longer terminate on the body but out in the field.

Figure 10 shows the final output, the initial value surface, derived by interpolation in the RSMoC-generated data. The data are presented "ring-wise," with each of the points being located on one of the specified meridian planes in the wind axis system. These data are also written on TAPE4, which can be defined as the system punch tape or a utility tape, for later reading by the the Three-Dimensional Method of Characteristics Program.

SECTION III

OPERATIONAL ASPECTS

1. CORE AND TIME REQUIREMENTS

The program requires approximately $100,000_8$ words of memory to load and execute on the CDC 6600 computer.

The program requires 30-40 seconds of Central Processor Unit (CPU) time to compile. A normal run will require 15-20 seconds CPU time for execution. Very little difference has been found between the execution times generated by the OPT = 1 and OPT = 2 compiler options.

2. ERROR MESSAGES

The built-in error messages in the program are listed below. They are arranged according to the OVERLAY segment and the program or subroutine in which they are generated. Probable causes and possible corrective measures are outlined.

However, no such discussion can be complete. The program is too complex to allow every eventuality to be foreseen. When failures do occur in the Blunt Body portion of the program, it is probably due to a poor choice in the number of steps to the body. If the failure is in the Rotationally Symmetric Method of Characteristics portion of the program, a slight change in the step size, the number of points on the initial value line, or M_{body} for the IVL will usually allow the program to continue. In the latter case, a plot of the characterisitc network will usually reveal the problem.

A failure in the Rotationally Symmetric Method of Characteristics portion of the program will result in the print option, INCASE, being reset to a value of 4 and the control being set back to repeat the calculation of the present characteristic. The physical appearance is the sudden onset of extremely detailed print. If the failure occurs in the region governed by the initial value line (since the program cannot rewind the input tape) the next point on the IVL is read in, and a detailed print of that calculation up to but not including the body point is given. By plotting the characteristic data, the troublesome point is easily spotted.

OVERLAY (BLUNT, 1, 0)

Program BLUNTS

COMPUTED FEWER THAN 3 BODY POINTS TWICE

Not enough shock points. Increase number of shock points.
Too many steps to the body. Decrease number of steps.

FAILED TO FIND A SUPERSONIC BODY POINT IN 4 TRIES

Too few shock points. Increase number of shock points.

FAILED IN G CONVERGENCE

No solution for the η-step multiplier. Change number of steps to the body

EXCEEDED 9 ITERATIONS FOR BODY

Convergence criterion too small. Relax convergence criterion.

Roughness in the relationship between B_{body} and B_{shock}. Plot the individual data, and establish a starting value for B_{shock}.

ZERO OR NEGATIVE G

Too many steps to the body. Decrease number of steps.

Subroutine BODY

NO ROOT FOR DELTA

Poorly defined body due to roughness and/or few body points. Increase number of points on the shock and/or decrease number of steps to the body.

INSUFFICIENT NUMBER OF BODY POINTS FOR CURVE FIT

Not enough body points. Increase number of shock points and/or decrease number of steps to the body.

Subroutine IDTHER

ROOT2 FAILURE, TR= _____ BM= _____ AM= _____

Calculated static pressure is greater than calculated total pressure. Arrange number of points on the shock and number of steps to the body so that $\Delta\eta$ is approximately equal to $\Delta\xi$

OVERLAY (BLUNT, 2, 0)

Program IVLS

DID NOT FIND SUPERSONIC POINT M= __

Not enough supersonic flow region in the Blunt Body portion of the program. Increase number of points along the shock.

Subroutine BBOUT

EXTRAPOLATED SONIC POINT

Not enough supersonic flow region in the Blunt Body portion of the program. Decrease number of steps to the body.

OVERLAY (BLUNT, 3, 0)

Subroutine BODI

X1 LIES BEYOND LAST SEGMENT

Probable error in body description and IVL geometry. Check input data

POINT LIES BEYOND END OF BODY X3= _____ XB= _____

Body description geometry is not sufficient for given data points. Plot characteristics to check for geometric incompatibilities, and check input data.

ITERATION FAILED

No convergence on body point. Plot each iteration and check for errors.

IM EXCEEDED 25

Body point Mach number is less than 1.0. Increase M_{body} in input data.

Subroutine FIELD

EXCEEDED 3*50 ITERATIONS IN FIELD POINT

Probable geometric inconsistency. Check input data, plot characteristics data.

FAILED TO CONVERGE S3 IN SPECIAL LOOP

Poor definition of the relationship between the stream function and total pressure. Check mass-entropy table, may be a read error by computer.

FAILED TO CONVERGE D3 IN SLOW LOOP

Geometric inconsistency. Check input data.

DR FAILURE IN FIELD

First approximation gave the correct answer, making the residual almost zero. The subsequent iteration was therefore thrown off. Change step size slightly.

FAILED IN FIELD, IMM= ___ P3= _______

Field point Mach number is less than 1.0. Plot characteristics data, check for geometric inconsistencies.

TRIED 10 TIMES TO CONVERGE POINT BUT STILL SUBSONIC

Field point Mach number is less than 1.0. Increase M_{body} in input data.

```
**NOTE****CONVERGENCE CRITERIA MULTIPLIED BY 10** , FOR POINT
```

Poor convergence at field point. Plot characteristics data to determine troublesome point.

```
OSCILLATION IN FIELD FOR DELS
```

Convergence problem in determining the local value of the stream function. Examine mass-entropy table for smoothness; may be a read error by computer.

Subroutine ENTROP

```
AFTER LAST ENTRY IN S/R TABLE. GMASS= ______,SR SET TO
LAST ENTRY, ______,W/CORRESPONDING MASS ______

BEFORE FIRST ENTRY IN S/R TABLE. GMASS= ______,SR SET TO
FIRST ENTRY, ______,W/CORRESPONDING MASS ______

BEFORE FIRST ENTRY IN S/R TABLE.   S/R= ______. GMASS SET TO
FIRST ENTRY, ______,W/CORRES. S/R= ______

BEYOND LAST ENTRY IN S/R TABLE.   S/R= ______ . GMASS SET TO
LAST ENTRY, ______,W/CORRES. S/R= ______
```

These statements are advisory in nature, indicating that the mass-entropy table and the local calculations are poorly matched. While the calculation will continue, the mass-entropy table, the geometric parameters, and the input data should be examined for errors.

Subroutine WAVE

```
FAILED TO CONVERGE SHOCK ANGLE IN 50 TRIES
```

The subroutine was given an incorrect deflection angle across the shock. Trace origin of the deflection angle, using check-out print.

Subroutine SHOCK

```
SM IS ZERO OR NEGATIVE IN RSHOCK
```

A negative step size is specified or complications arising from the mesh build-up resulted in a negative step size. Check input data and/or plot characteristics data.

```
OSCILLATED THROUGH NGATE THEN IGATE

OSCILLATED THROUGH IGATE THEN NGATE
```

Both of the above messages indicate that base point data for the shock point solution are being determined by extrapolation rather than interpolation. Plot the progress of the shock point solution to find out where the incorrect extrapolation occurs.

EXCEEDED 50 ITERATIONS FOR POINT 4

Geometric inconsistencies in the mesh build-up. Plot characteristic mesh to locate trouble spot.

EXCEEDED 75 ITERATIONS FOR SHOCK

Poor initial data. Plot characteristic mesh to locate problem area.

SECTION IV

LOGICAL STRUCTURE

1. OVERLAY STRUCTURE, READ/WRITE LOGIC

The logic of the program is controlled by the master program and uses four first level OVERLAY segments. The master program consists of a main program to call each OVERLAY segment and one general use subroutine. The Blunt Body Program, the Initial Value Line Interpolation Program, the Rotationally Symmetric Method of Characteristics Program, and the Initial Value Surface Interpolation Program each comprise one OVERLAY segment. Figure 11 shows the OVERLAY structure and the storage requirements.

Figure 12 shows the read/write logic structure. TAPE2 and TAPE3 would be the same device if minor changes were made in the program logic. TAPE4 is shown as a dotted output device to indicate that this could be either the system punch file or a scratch tape, as the user specifies.

2. INTERDEPENDENCE OF SUBROUTINES

The Calling-Called matrices for three of the four OVERLAY segments are shown in Figure 13. The segment comprised of the Initial Value Surface Program consists of one main program which in turn calls two general purpose subroutines. This matrix, due to its simplicity, is not shown in Figure 13.

3. LISTING

The complete listing of the IVS program is given below, following Figure 13.

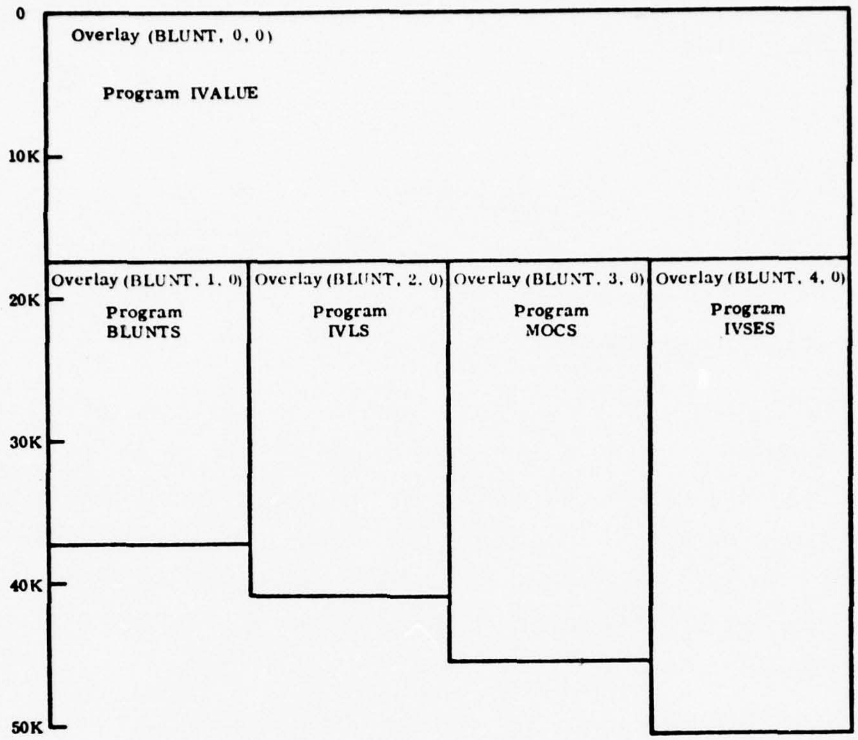


FIGURE 11. OVERLAY STRUCTURE

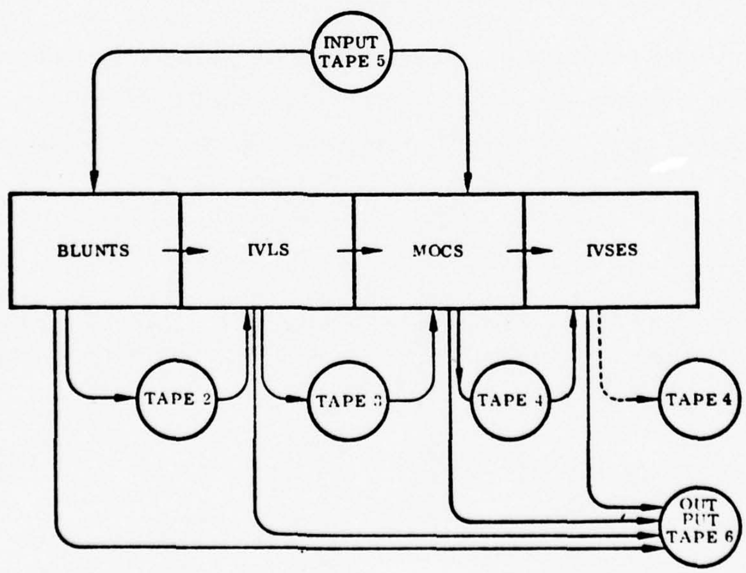


FIGURE 12. READ/WRITE LOGIC

Calling \ Called	BODY	XY	DIFATE	IDTHER	IMAIN	ENTRPG
BLUNTS					●	
IMAIN	●	●	●	●		●

(a) Program BLUNTS

Calling \ Called	DETER	BBOUT	ENTRPG
IVLS	●	●	●
DETER			●
BBOUT			●

(b) Program IVLS

Calling \ Called	BODI	OPER8	FIELD	VMACH	ENTROP	TSON	DIF	CHANGE	INPUT	SHAPE	DROP	OUTPUT	WAVE	SHOCK	MESH	PAGE	ANGLE	VELOC	ENTRPG
MOCS	●	●			●	●		●	●	●	●	●	●	●	●	●	●	●	
BODI				●												●			
OPER8			●																
FIELD				●	●												●		
ENTROP																●			●
SHAPE																●			
DROP								●											
OUTPUT																●			
WAVE						●											●		
SHOCK					●		●				●		●				●		

(c) Program MOCS

FIGURE 13. CALLING-CALLED MATRIX FOR IVS PROGRAM

```
      OVERLAY(BLUNT,0,0)
      PROGRAM IVALUE(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1 TAPE7=PUNCH,TAPE3,TAPE4,TAPE2)
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,AC,ACHM,GAMA,ALFA,ELAM,RINGS
     1,CPR(88),ONE05,M,BDRAG
      COMMON /TRANS/ L0,L1,LL,MS
C       INITIAL VALUE SURFACE PROGRAM
      CALL FTNBIN (1,0,DUMMY)
1     REWIND 3
      CALL OVERLAY (5HBLUNT,1,0)
      CALL OVERLAY (5HBLUNT,2,0)
      CALL OVERLAY (5HBLUNT,3,0)
      CALL OVERLAY (5HBLUNT,4,0)
      GO TO 1
      END
```

```
      SUBROUTINE ENTRPG (X1,X2,X3,Y1,Y2,Y3,X,Y)
C
C     ******DEVELOPED BY AERODYNAMICS RESEARCH BRANCH, NORAIR***********
      A=X-X1
      B=X-X2
      C=X-X3
      D=X1-X2
      E=X1-X3
      G=X2-X3
      AB=A*B
      AC=A*C
      BC=B*C
      R=ABS(D/G)
      IF (R-1.0E-02) 2,2,1
1     IF (R-1.0E+02) 8,2,2
2     IF (AB) 3,3,4
3     DY=Y1-Y2
      DX=D
      DD=A
      CON=Y1
      GO TO 6
4     IF (BC) 5,5,7
5     DY=Y3-Y2
      DX=-G
      DD=C
      CON=Y3
6     Y=CON+DD*DY/DX
      GO TO 9
7     IF (ABS(A)-ABS(C)) 3,3,4
8     F1=(Y1*BC)/(D*E)
      F2=(Y2*AC)/(-D*G)
      F3=(Y3*A*B)/(E*G)
      Y=F1+F2+F3
9     RETURN
      END
```

```
      OVERLAY (BLUNT,1,0)
      PROGRAM BLUNTS
C
      COMMON /ALPHA/ X,R,PTOPI,CPI,DETA,ETASTG,XISON,ETASON,GG(15),ACHM2
     1,DEM,XXB(88),YYB(88),ACH(100),STAG,QOOOFL(10),DETA02
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,B,ACHM,GAMA,ALFA,ELAM,RINGS,
     1CPR(88),ONEO5,M
      COMMON /KLAPHA/ IBY,NETA,KPRNT,NB,KDIM,KOM(10),MP
      DIMENSION XA(9),HOL(7)
      REWIND 2
      REWIND 3
      READ (5,45) HOL
      IF(EOF(5)) 56,57
56    CALL EXIT
57    CONTINUE
      DO 1 I=1,7
1     IF (HOL(I).EQ.10H          ) HOL(I)=10H* * * * *
      WRITE (6,55) HOL
      READ (5,44) ACHM,GAMA,ALFA,ELAM,RINGS,ONEO5
      READ (5,44) BBW,BTEST,STEPS,B,POINTS,EM
      WRITE (6,51) ACHM,GAMA,ALFA,ELAM,RINGS,ONEO5,BBW,BTEST,STEPS,B,
     1POINTS,EM
      M=EM
      DEM=1
      GG(1)=GAMA
      GG(2)=(GAMA-1.0)*0.5
      GG(3)=GAMA/(GAMA-1.0)
      GG(4)=1.0-1.0/GAMA
      GG(5)=GAMA+1.0
      GG(6)=GAMA-1.0
      GG(7)=1.0/GAMA
      GG(8)=2.0/(GAMA-1.0)
      GG(9)=1.0/(GAMA-1.0)
      GG(10)=2.0*GG(1)
      GG(11)=4.0*GG(1)
      ACHM2=ACHM**2
      PTOPI=(1.0+GG(2)*(ACHM**2))**GG(3)
      CPI=2.0/(GAMA*(ACHM**2))

      B2=0.0
```

```
      BI=0.0
      BOD1=0.0
      BOD2=0.0
      IBY=0
      IF (ACHM) 2,2,3
2     ACHM=ABS(ACHM)
      MP=4
      IBY=1
      GO TO 4
3     MP=0
4     NY=POINTS
      D1=ACHM**2-1.0
      DEM=1.0
      KDIM=DEM
      KOM(1)=1.0+DEM
      Q000FL(1)=1.0/(1.0+DEM)
      KOM(2)=KDIM-1
      KOM(3)=2*KDIM
      KOM(4)=2/(1+KDIM)
      Q000FL(3)=KOM(1)
      Q000FL(4)=GG(11)*ACHM2
      Q000FL(5)=GG(5)*ACHM2
      Q000FL(6)=1.0+GG(2)*ACHM2
      IF (B) 7,5,7
5     B=0.5-0.559*((ACHM-1.0)**(-1.19))
6     DY=0.02
      DETA=(.02+(.0576/(ACHM**2)))
      G=1.0
      GO TO 8
7     READ (5,44) G,DETA,DY
      IF (G) 6,6,8
8     JLOOP=0
      DETB=DETA
      KPRNT=MP
      ILOOP=0
9     NY=POINTS
      IF (B-(-1.0/D1)) 12,10,10
10    XSOUND=SQRT(D1/(B*D1+1.0))
      IF (XSOUND-(FLOAT(NY)-0.5)*DY) 11,12,12
11    DY=XSOUND/POINTS
```

```
12    WRITE (6,54)
      WRITE (6,53) ACHM,GAMA
      CALL IMAIN (NY)
      NB=NB
      ILOOP=ILOOP+1
      IF (ABS(BBW-BEE)-BTEST) 36,36,13
13    DELETA=(1.0-ETASTG)*0.98
      IF (NB-3) 14,14,16
14    G=1.0
      DETA=DELETA/STEPS
      IF (ILOOP-1) 9,9,15
15    WRITE (6,46)
      WRITE (6,47) (XXB(I),YYB(I),I=1,NB)
      GO TO 25
16    IF (ILOOP-1) 21,21,17
17    IF (ACH(NB)-1.0) 18,18,20
18    IF (ILOOP-4) 21,21,19
19    WRITE (6,48)
      ILOOP=1
      GO TO 34
20    IF (ACH(NB-1)-1.0) 34,21,21
21    DELETB=(1.0-DELETA-ETASON)*1.3
      AM=STEPS+1.0
      R=DELETB/DELETA
      IG=0
      GP=STEPS*(1.0+R)/AM
      IF (G-1.0) 26,22,26
22    IF (GP-1.0) 23,23,28
23    G=1.0-2.0*(1.0-GP)
      IF (G) 24,24,29
24    WRITE (6,52) G,GP
25    CALL EXIT
26    IF (GP-1.0) 27,27,28
27    G=0.7*GP
      GO TO 29
28    G=1.3*GP
29    T2=G**AM
      IG=IG+1
      IF (IG-50) 31,31,30
30    WRITE (6,49) G,F,FP,DELETA,DELETB
```

```
      GO TO 25
31    F=T2-(T2*(1.0+R))/G+R
      FP=(AM*T2-(STEPS*T2/G)*(1.0+R))/G
      RATF=F/FP
      IF (ABS(RATF)-.00001) 33,33,32
32    G=G-RATF
      GO TO 29
33    DETA=DELETA*(G-1.0)/(T2/G-1.0)
      DETB=DETA
      GO TO 9
34    JLOOP=JLOOP+1
      IF (JLOOP-9) 38,38,35
35    WRITE (6,50)
      GO TO 25
36    DO 37 I=1,9
37    XA(I)=0.0
      WRITE (2) (XA(I),I=1,6)
      WRITE (2) (XA(I),I=1,6)
      GO TO 43
38    B2=B1
      B1=B
      BOD2=BOD1
      BOD1=BEE
      REWIND 2
      IF (JLOOP-2) 39,40,40
39    SIG=SIGN(1.0,B)
      B=B*(BBW/BEE)**(SIG)
      GO TO 21
40    B=(B2-B1)/(BOD2-BOD1)
      IF (B) 41,41,42
41    B=B1+(B1-B2)
      B1=B2
      BOD1=BOD2
      GO TO 21
42    B=B*(BBW-BOD2)+B2
      GO TO 21
C
43    BEE=BBW
C
44    FORMAT (6E12.4)
```

```
45    FORMAT (7A10)
46    FORMAT (1X39HCOMPUTED FEWER THAN 3 BODY POINTS TWICE,80(1H*))
47    FORMAT (//9X6HX-BODY,9X6HY-BODY/(2E15.6))
48    FORMAT (1X,49HFAILED TO FIND A SUPERSONIC BODY POINT IN 4 TRIES)
49    FORMAT (1X23HFAILED IN G CONVERGENCE/1X2HG=E15.6/1X2HF=E15.6/1X3HF
     1P=E15.6/1X7HDELETA=E15.6/1X7HDELETB=E15.6)
50    FORMAT (1X30HEXCEEDED 9 ITERATIONS FOR BODY)
51    FORMAT (1H0,T30,5H*****/T30,15H* I *  MACH NO.,T60,1H=,G12.5/T30,
     112H* N *  GAMMA,T60,1H=,G12.5/T30,22H* P *  ANGLE OF ATTACK,T60,1H
     2=,G12.5,T73,4HDEG./T30,17H* U *  BODY SLOPE,T60,1H=,G12.5,T73,4HDE
     3G./T30,19H* T *  NO. OF RINGS,T60,1H=,G12.5/T30,29H*   *  INTERPOL
     4ATION MACH NO.,T60,1H=,G12.5/T30,21H* D *  BODY BLUNTNESS,T60,1H=,
     5G12.5/T30,14H* A *  BLUNTNE,12HSS TOLERANCE,T60,1H=,G12.5/T30,12H*
     6 T *  STEPS,T60,1H=,G12.5/T30,15H* A *  B(SHOCK),T60,1H=,E16.7/T30
     7,20H* : *  NO. OF POINTS,T60,1H=,G12.5/T30,23H*****  NO. OF MERIDI
     8ANS,T60,1H=,G12.5)
52    FORMAT (10X18HZERO OR NEGATIVE G,10(6H***    ))
53    FORMAT (25X22HFREE STREAM CONDITIONS/15X2HM=E15.6/15X6HGAMMA=
     10PF10.6)
54    FORMAT (1H1)
55    FORMAT (2H1*,14(2H *),7A10,14(2H* ))
      END
```

```
      SUBROUTINE BODY (NBDY)
      COMMON /ALPHA/ X,R,PTOPI,CPI,DETA,ETASTG,XISON,ETASON,GG(15),ACHM2
     1,DEM,XXB(88),YYB(88),ACH(100),STAG,Q000FL(10),DETAO2
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,B,ACHM,GAMA,ALFA,ELAM,RINGS,
     1CPR(88),ONEOS,MDM
      COMMON /KLAPHA/ IBY,NETA,KPRNT,NB,KDIM,KOM(10),MP
      DIMENSION WYCAL(88)
      N=NB
      GN=N-3
      IF (GN) 1,1,2
    1 WRITE (6,30)
      ETASTG=YYB(1)*2.0/DY
      GO TO 27
    2 T1=N
      T2=0.0
      T3=0.0
      T4=0.0
      T5=0.0
      S1=0.0
      S2=0.0
      S3=0.0
      DO 3 I=1,N
      T2=T2+XXB(I)
      TT=XXB(I)*XXB(I)
      T3=T3+TT
      TT=TT*XXB(I)
      T4=T4+TT
      TT=TT*XXB(I)
      T5=T5+TT
      TT=YYB(I)*YYB(I)
      S1=S1+TT
      TT=TT*XXB(I)
      S2=S2+TT
      TT=TT*XXB(I)
    3 S3=S3+TT
      DENOM=T1*(T3*T5-T4*T4)+T2*(2.0*T3*T4-T2*T5)-T3*T3*T3
      U1=S1*(T3*T5-T4*T4)+S2*(T3*T4-T2*T5)+S3*(T2*T4-T3*T3)
      U2=S1*(T3*T4-T2*T5)+S2*(T1*T5-T3*T3)+S3*(T2*T3-T1*T4)
      U3=S1*(T2*T4-T3*T3)+S2*(T2*T3-T1*T4)+S3*(T1*T3-T2*T2)
      ALPHA=U1/DENOM
```

```
      BETA=U2/DENOM
      GAMMA=U3/DENOM
      T1=-0.5*BETA/GAMMA
      T2=ALPHA/GAMMA
      T2=T1*T1-T2
      IF (T2) 4,4,5
4     DELCHK=XXB(1)
      GO TO 12
5     T2=SQRT(T2)
      DELP=T1+T2
      DELM=T1-T2
      IF (DELP) 9,9,6
6     IF (DELM) 8,8,7
7     DELCHK=AMIN1(DELM,DELP)
      GO TO 12
8     DELCHK=DELP
      GO TO 12
9     IF (DELM) 10,10,11
10    WRITE (6,29) DELP,DELM
      RETURN
11    DELCHK=DELM
12    BEE=-GAMMA
      ARBE=0.5*BETA-BEE*DELCHK
      CARD=DELCHK
      DO 15 J=1,N
      WYCAL(J)=(2.0*ARBE*(XXB(J)-DELCHK)-BEE*(XXB(J)-DELCHK)**2)
      IF (WYCAL(J)) 13,14,14
13    WYCAL(J)=0.0
      GO TO 15
14    WYCAL(J)=SQRT(WYCAL(J))
15    CONTINUE
      WRITE (6,28) DELCHK,BEE,ARBE
      NDRAG=NBDY
      CMP=(1.0+DEM)/(ARBE**(1+KDIM))
      CD=0.5*CPR(1)*(YYB(1)**KDIM)*CMP
      CD1=CD
      DO 16 J=2,NDRAG
      CD=0.5*(CPR(J)+CPR(J-1))*((YYB(J)-YYB(J-1))**(KDIM))*CMP+CD
16    CPR(J-1)=CD
      DO 17 J=1,NDRAG
```

```
      N1=NDRAG-J+1
17    CPR(N1)=CPR(N1-1)
      CPR(1)=CD1
      SXI2E=0.0
      SXI=0.0
      SXI2=0.0
      SXI3=0.0
      SXI4=0.0
      SETA=0.0
      SETA2=0.0
      SXIM=0.0
      SXI2M=0.0
      XI=-0.5*DY
      WRITE (6,31)
      DO 18 J=1,N
      XI=XI+DY
      ETAB=YYB(J)/XI
      WRITE (6,32) J,ETAB,XI,ACH(J)
      SXI=SXI+XI
      SXIM=SXIM+XI*ACH(J)
      XI2=XI*XI
      SXI2=SXI2+XI2
      SXI2M=SXI2M+XI2*ACH(J)
      SXI2E=SXI2E+ETAB*XI2
      XI2=XI2*XI
      SXI3=SXI3+XI2
      XI2=XI2*XI
      SXI4=SXI4+XI2
      SETA=SETA+ETAB
      ETAB=ETAB*ETAB
18    SETA2=SETA2+ETAB
      BN=N
C     BODY EQUATION
      D1=BN*SETA2-(SETA)**2
      A1=(BN*SXI2E-SETA*SXI2)/D1
      B1=(SETA*SXI2E-SETA2*SXI2)/D1
C     MACH NUMBER EQUATION
      D2=SXI2*SXI4-SXI3**2
      A2=(SXIM*SXI4-SXI2M*SXI3)/D2
      B2=(SXI2M*SXI2-SXIM*SXI3)/D2
```

```
      ETASTG=B1/A1
      IF (B2) 20,19,20
19    XISON=1.005/A2
      GO TO 23
20    XISON=(A2**2+4.0*B2*1.005)
      IF (XISON) 21,21,22
21    XISON=1.005/A2
      GO TO 23
22    XISON=(-A2+SQRT(XISON))/(2.0*B2)
23    ETASON=(XISON**2+B1)/A1
      IF (B) 25,24,25
24    XSTAG=0.5*(1.0-ETASTG**2)
      GO TO 26
25    XSTAG=(1.-SQRT(1.-B+B*(ETASTG**2)))/B
26    STAG=XSTAG
27    RETURN
C
C
28    FORMAT (1X119(1H*)/1X119(1H*)//5X6HDELTA=E15.7/3X8HB(BODY)=E15.7/
     13X8HR(BODY)=E15.7/1X119(1H*)/1X119(1H*))
29    FORMAT (5X23HNO ROOT FOR DELTA,DELM=E15.7,5X5HDELP=E15.7)
30    FORMAT (5X48HINSUFFICIENT NUMBER OF BODY POINTS FOR CURVE FIT)
31    FORMAT (10X3HNO.,15X3HETA,15X2HXI,15X4HMACH)
32    FORMAT (I12,E25.6,3E18.6)
      END
```

```
      SUBROUTINE XY (XI,ETA,B,X,R,FA,SQROOT)
      SQROOT=SQRT(1.0-B+B*ETA**2)
      FA=SQRT(1.-B*(XI**2))
      IF (B) 2,1,2
1     X=0.5*(1.0+XI**2-ETA**2)
      GO TO 3
2     X=(1.0-FA*SQROOT)/B
3     R=ETA*XI
      RETURN
      END
```

```
      SUBROUTINE DIFATE (NX,DX,YBLK,DFX,D2FX)
      DIMENSION YBLK(100),FBLK(9),DEL1(6),DEL2(5),DEL3(4),DEL4(3),DEL5(2
     1)
      IF (NX-4) 3,1,1
1     DO 2 I=1,6
      NXI=NX-4+I
2     DEL1(I)=YBLK(NXI+1)-YBLK(NXI)
      GO TO 6
3     DO 4 I=1,6
4     FBLK(I+3)=YBLK(I)
      FBLK(1)=YBLK(3)
      FBLK(2)=YBLK(2)
      FBLK(3)=YBLK(1)
      DO 5 I=1,6
      NXI=NX+I-1
5     DEL1(I)=FBLK(NXI+1)-FBLK(NXI)
6     DO 7 I=1,5
7     DEL2(I)=DEL1(I+1)-DEL1(I)
      DO 8 I=1,4
8     DEL3(I)=DEL2(I+1)-DEL2(I)
      DO 9 I=1,3
9     DEL4(I)=DEL3(I+1)-DEL3(I)
      DO 10 I=1,2
10    DEL5(I)=DEL4(I+1)-DEL4(I)
      DEL6=DEL5(2)-DEL5(1)
      BIG1=AMAX1(ABS(DEL1(1)),ABS(DEL1(2)),ABS(DEL1(3)),ABS(DEL1(4)),ABS
     1(DEL1(5)),ABS(DEL1(6)))
      BIG2=AMAX1(ABS(DEL2(1)),ABS(DEL2(2)),ABS(DEL2(3)),ABS(DEL2(4)),ABS
     1(DEL2(5)))
      BIG3=AMAX1(ABS(DEL3(1)),ABS(DEL3(2)),ABS(DEL3(3)),ABS(DEL3(4)))
      BIG4=AMAX1(ABS(DEL4(1)),ABS(DEL4(2)),ABS(DEL4(3)))
      BIG5=AMAX1(ABS(DEL5(1)),ABS(DEL5(2)))
      DFX=(DEL1(3)+DEL1(4))/(2.0*DX)
      IF (BIG2-BIG1) 12,11,11
11    D2FX=0.0
      GO TO 16
12    D2X=DX**2
      D2FX=DEL2(3)/D2X
      IF (BIG3-BIG2) 13,16,16
13    DFX=DFX-(DEL3(2)+DEL3(3))/(12.0*DX)
```

```
      IF (BIG4-BIG3) 14,16,16
14    D2FX=D2FX-(DEL4(2)/(12.0*D2X))
      IF (BIG5-BIG4) 15,16,16
15    DFX=DFX+(DEL5(1)+DEL5(2))/(60.0*DX)
      D2FX=D2FX+(DEL6/(90.0*D2X))
16    RETURN
      END
```

```
      SUBROUTINE IDTHER (YA,E,F,P,AE,T,SY,SE,BM,FA,SQROOT)
      COMMON /ALPHA/ X,R,PTOPI,CPI,DETA,ETASTG,XISON,ETASON,GG(15),ACHM2
     1,DEM,XXB(88),YYB(88),ACH(100),STAG,QOOOFL(10),DETA02
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,B,ACHM,GAMA,ALFA,ELAM,RINGS,
     1CPR(88),ONE05,MDM
      COMMON /KLAPHA/ IBY,NETA,KPRNT,NB,KDIM,KOM(10),MP
      NETA=NETA
      PR=GG(1)*P*ACHM2
      PTR=(GG(1)*ACHM2*F)**(-GG(9))
      TERM=PR/(PTOPI*PTR)
      IF (TERM-1.0) 2,2,1
1     WRITE (6,16) PR,PTR,TERM
      TERM=1.0
      BM=0.0
      GO TO 3
2     TERM=TERM**(-GG(4))
      BM=SQRT((TERM-1.0)*GG(8))
3     TR=(1.0+GG(2)*ACHM2)/(1.0+GG(2)*BM**2)
      ROOT2=TR*(BM/ACHM)**2
      IF (ROOT2) 4,5,5
4     WRITE (6,15) TR,BM,ACHM
5     VR=SQRT(ROOT2)
      TD=57.29578*T
      IF (E-1.0) 7,6,6
6     DEL=T-ATAN(SIN(T)/(COS(T)*AE))
      GO TO 8
7     DEL=ATAN((E/YA)*(FA/SQROOT))+ATAN((-SY/SE)*(FA/SQROOT))
8     DELD=57.29578*DEL
      WRITE (2) X,R,BM,PTR,PR,DELD
      IF (E-1.0) 10,9,9
9     WRITE (2) TD
10    IF (IBY) 13,13,11
11    WRITE (6,17) X,R,BM,VR,PTR,PR,TR,AE,DELD
      IF (E-1.0) 13,12,12
12    WRITE (6,14) TD
13    RETURN
C
C
14    FORMAT (1H+,110X,F9.5)
15    FORMAT (5X,17HROOT2 FAILURE,TR=E14.6,5X,3HBM=E14.6,5X,3HAM=E14.6)
```

```
16     FORMAT (5X3HPR=E15.6/10X4HPTR=E15.6/15X5HTERM=E15.6)             G  41
17     FORMAT (2E13.6,0PF11.7,5E13.6,0P2F9.5)                            G  42
       END                                                               G  43
```

```
      SUBROUTINE IMAIN (NYY)
C                INITIAL VALUE PROGRAM FOR 3-D MOC
      COMMON /ALPHA/ X,R,PTOPI,CPI,DETA,ETASTG,XISON,ETASON,GG(15),ACHM2
     1,DEM,XXB(88),YYB(88),ACH(100),STAG,Q000FL(10),DETAO2
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,B,ACHM,GAMA,ALFA,ELAM,RINGS,
     1CPR(88),ONE05,MOM
      COMMON /KLAPHA/ IBY,NETA,KPRNT,NB,KDIM,KOM(10),MP
      DIMENSION PR(100),PB(100),DEN(100),FNS(100),TH(100),STRM(100),
     1DSTRY(100),D2STRY(100),DSTRN(100),D2STNY(100),D2STRN(100),DPDN(100
     2),DPDY(100),DSORDS(100),DSORDP(100),DSORDD(100),SAV2(100),PBXY(3,
     388),FNXY(3,88),DSYXY(3,88),DSNXY(3,88),STRXY(3,88)
      DATA CPR/88*0.0/
      REWIND 2
      Q000FL(2)=1.0-B
      DETAO2=DETA*0.5
      NBDY=0 $FOO=0.0
      NY=NYY
      WRITE (6,89) B,DETA,DY,G
      NB=0
      NETA=1
      ETA=1.0
      ETA1=1.0
      Y=0.5*DY
      ACHL=ACHM
      LINES=16
      IF (KPRNT-1) 2,1,1
    1 WRITE (6,71)
      WRITE (6,82)
    2 DO 8 I=1,NY
      TH(I)=ATAN(SQRT(1.0-(B*Y**2))/Y)
      CALL XY (Y,ETA,B,X,R,FA,SQ)
      ACHN=ACHL*SIN(TH(I))
      ACHN2=ACHN**2
      PR(I)=((2.0*ACHN2)-GG(4))/Q000FL(5)
      PB(I)=PR(I)
      DEN(I)=(GG(5)*ACHN2)/(GG(6)*ACHN2+2.0)
      DRT=(Q000FL(6)/(1.0+GG(2)*(ACHL**2)))**GG(9)
      DEN(I)=DEN(I)*DRT
      FNS(I)=PR(I)/(DEN(I)**GG(1))
      CALL IDTHER (Y,ETA,FNS(I),PR(I),DEN(I),TH(I),0.0,0.0,ACH(I),FA,SQ)
```

```
      STRM(I)=(Y**KOM(1))/Q000FL(3)
      DSTRY(I)=Y**KDIM
      D2STRY(I)=DEM*Y**(KOM(2))
      DSTRN(I)=DEN(I)*(Y**KOM(1))
      IF (I-88) 3,3,4
3     FNXY(1,I)=FNS(I)
      PBXY(1,I)=PB(I)
      DSYXY(1,I)=DSTRY(I)
      DSNXY(1,I)=DSTRN(I)
      STRXY(1,I)=STRM(I)
4     IF (KPRNT-1) 8,5,5
5     IF (LINES-45) 7,7,6
6     LINES=1
      WRITE (6,84)
      WRITE (6,82)
7     LINES=LINES+1
8     Y=Y+DY
      WRITE (6,70) NETA,ETA,NY
      X=0.0
      ACHL=ACHM
      ACHL=ACHL**2
      PBST=((2.0*ACHL)-GG(4))/(GG(5)*ACHL)
      DENST=(GG(5)*ACHL)/(GG(6)*ACHL+2.0)
      FNST=PBST/(DENST**GG(1))
      IF (KPRNT-3) 10,9,9
9     WRITE (6,72) ETA
      WRITE (6,85)
10    Y=0.5*DY
      IF (KPRNT-3) 13,11,11
11    DO 12 I=1,NY
      WRITE (6,69) Y,STRM(I),DSTRY(I),D2STRY(I),DSTRN(I)
12    Y=Y+DY
13    NTR=1
14    IF (KPRNT-3) 16,15,15
15    WRITE (6,73) ETA
      WRITE (6,86)
16    Y=0.5*DY
      DO 18 I=1,NY
      SSQ=Q000FL(3)*STRM(I)
      SSQ2=SSQ**(2.0*Q000FL(1))
```

```
      F1=1.0+(Q000FL(2))*SSQ2
      F2=(ACHM2)*(1.0-B*SSQ2)
      F3=Q000FL(4)/(GG(6)*(SSQ**(KOM(2))))
      F4=(1.0/F2)*(1.0/(2.0*F1+GG(6)*F2))
      F5=(1.0/(F1*(GG(10)*F2-GG(6)*F1)))
      DSORDS(I)=F3*(F4-F5)
      DSORDP(I)=GG(9)/PR(I)
      DSORDD(I)=-(GG(3)/DEN(I))
      IF (KPRNT-3) 18,17,17
17    WRITE (6,74) Y,SSQ,F1,F2,F3,F4,F5,DSORDS(I),DSORDP(I),DSORDD(I)
18    Y=Y+DY
      NY3=NY-3
      DO 19 I=1,NY3
      CALL DIFATE (I,DY,DSTRN,D2STNY(I),D2)
      CALL DIFATE (I,DY,PR,DPDY(I),D2)
19    CONTINUE
      IF (KPRNT-3) 21,20,20
20    WRITE (6,75) (D2STNY(I),I=1,NY3)
      WRITE (6,76) (DPDY(I),I=1,NY3)
      WRITE (6,77) ETA
      WRITE (6,87)
21    Y=0.5*DY
      AD=1.0-B
      D7=AD+(B*ETA**2)
      D16=ETA**2
      D17=DEM/ETA
      D18=-B*ETA
      DO 24 I=1,NY3
      YSQ=Y**2
      D1=1.0-B*YSQ
      D2=(Y*ETA)**(KOM(3))
      D3=(AD*YSQ)+D16
      DSTRY2=DSTRY(I)**2
      DSDN=DSORDD(I)*DEN(I)
      D1OD2=D1/D2
      D4=(DEN(I)*D3)+(D1OD2)*(DSTRY2)*(DSORDP(I)/DSDN)
      IF (D4) 25,25,22
22    D5=(D1OD2)*(DSORDS(I)*(DSTRY2)*DSTRN(I))/DSDN
      ETAOD3=ETA/D3
      D6=D1OD2*DSTRY2*((ETAOD3)+(D17)-(D2STNY(I)/DSTRY(I)))
```

```
      DSTRN2=DSTRN(I)**2
      D8=ETAOD3*D7*DSTRN2/D2
      D9=(DSORDS(I)*DSTRY(I)-DSORDP(I)*DPDY(I))/DSDN
      DIMOY=DEM/Y
      D10=D10D2*(DSTRY(I)*DSTRN(I))*((B*Y/D1)+DIMOY-(D2STRY(I)/DSTRY(I))
     1+D9)
      DPDN(I)=(D5+D6+D8-D10)/D4
      CYOD3=AD*Y/D3
      D11=(D7/D2)*DSTRN2*(-(CYOD3)-DIMOY-D9+(D2STNY(I)/DSTRN(I)))
      D12=CYOD3*D1*DSTRY2/D2
      D13=DEN(I)*D3*DPDY(I)
      D14=(D2/(D7*DSTRY(I)))*(D11-D12+D13)
      D15=((DSORDS(I)*DSTRN(I))-(DPDN(I)*DSORDP(I)))/DEN(I)
      D2STRN(I)=D14+DSTRN(I)*(((D18)/D7)+D17+(D15/DSORDD(I)))
      IF (KPRNT-3) 24,23,23
23    WRITE (6,68) Y,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1DPDN(I),D2STRN(I)
24    Y=Y+DY
      GO TO (26,52), NTR
25    NY3=I-1
      NY=NY3+3
      GO TO (26,52), NTR
26    IF (ETA-1.0) 62,27,27
27    ETA=ETA-DETA
      Y=0.5*DY
      NY=NY-3
      NETA=NETA+1
      IF (KPRNT-3) 29,28,28
28    WRITE (6,67)
      WRITE (6,88)
29    DO 32 I=1,NY
      SAV1=DSTRN(I)
      SAV2(I)=DPDN(I)
      DSTRN(I)=DSTRN(I)-(DETA*D2STRN(I))
      STRM(I)=STRM(I)-DETAO2*(SAV1+DSTRN(I))
      PR(I)=PR(I)-(DETA*DPDN(I))
      IF (PR(I)) 33,33,30
30    SSQ=(Q000FL(3)*STRM(I))**(KOM(4))
      F1=1.0+Q000FL(2)*SSQ
      F2=ACHM2*(1.0-B*SSQ)
```

```
      F3=(2.0*F2-F1*GG(4))/(F1*Q000FL(5))
      F4=(2.0*F1+GG(6)*F2)/(GG(5)*F2)
      FNS(I)=F3*(F4**GG(1))
      DEN(I)=(PR(I)/FNS(I))**GG(7)
      IF (KPRNT-3) 32,31,31
31    WRITE (6,66) Y,SAV1,SAV2(I),DSTRN(I),STRM(I),PR(I),SSQ,F1,F2,F3,F4
     1,FNS(I),DEN(I)
32    Y=Y+DY
      GO TO 34
33    NY=I-2
34    IF (NETA-4) 48,35,35
35    Y=0.5*DY
      IF (KPRNT-2) 37,36,36
36    WRITE (6,84)
37    IKP=KPRNT
      KPRNT=KPRNT+2
      IBZ=IBY
      IBY=MP
      DO 44 I=1,NY
      IF (STRM(I)) 38,40,44
38    IF (STRXY(1,I)) 44,44,39
39    CALL ENTRPG (STRXY(1,I),STRXY(2,I),STRXY(3,I),ETA1,ETA2,ETA3,0.0,
     1ETAE)
      CALL ENTRPG (ETA1,ETA2,ETA3,PBXY(1,I),PBXY(2,I),PBXY(3,I),ETAE,PBE
     1)
      FNE=FNST
      DENE=(PBE/FNE)**GG(7)
      CALL ENTRPG (ETA1,ETA2,ETA3,DSYXY(1,I),DSYXY(2,I),DSYXY(3,I),ETAE,
     1DSYE)
      CALL ENTRPG (ETA1,ETA2,ETA3,DSNXY(1,I),DSNXY(2,I),DSNXY(3,I),ETAE,
     1DSNE)
      CALL XY (Y,ETAE,B,X,R,FA,SQROOT)
40    CONTINUE
      CALL IOTHER (Y,ETAE,FNE,PBE,DENE,TH(I),DSYE,DSNE,ACHE,FA,SQROOT)
      ACH(I)=ACHE
      CPR(I)=(2.0*PBE-CPI)
      IF (ACHE-1.0) 41,41,43
41    IF (I-NB-1) 42,42,43
42    NB=I
43    XXB(I)=X
```

```
      YYB(I)=R
      NBDY=I
   44 Y=Y+DY
      IBY=IBZ
      DO 45 I=1,NBDY
   45 CPR(I)=CPR(I)*YYB(I)
      KPRNT=IKP
      IF (NB) 48,48,46
   46 NBI=NB+2
      DO 47 I=NBI,NY
   47 ACH(I)=0.0
      CALL BODY (NBDY)
      NB=NB+1
      GO TO 65
   48 NY3=NY-3
      DO 49 I=1,NY3
      CALL DIFATE (I,DY,STRM,DSTRY(I),D2STRY(I))
   49 CONTINUE
      IF (KPRNT-3) 51,50,50
   50 WRITE (6,79) (DSTRY(I),I=1,NY3)
      WRITE (6,80) (D2STRY(I),I=1,NY3)
   51 NTR=2
      GO TO 14
   52 IF (KPRNT-2) 54,53,53
   53 WRITE (6,81) ETA
      WRITE (6,78)
      WRITE (6,83)
   54 Y=0.5*DY
      DO 59 I=1,NY3
      PB(I)=PB(I)-DETAO2*(SAV2(I)+DPDN(I))
      IF (PB(I)) 61,61,55
   55 DEN(I)=(PB(I)/FNS(I))**GG(7)
      CALL XY (Y,ETA,B,X,R,FA,SQROOT)
      CALL IDTHER (Y,ETA,FNS(I),PB(I),DEN(I),FOO,DSTRY(I),DSTRN(I),ACH(I
     1),FA,SQROOT)
      IF (KPRNT-2) 59,56,56
   56 IF (LINES-33) 58,58,57
   57 LINES=1
      WRITE (6,84)
      WRITE (6,83)
```

```
58    LINES=LINES+1
59    Y=Y+DY
      NY=NY-3
60    WRITE (6,70) NETA,ETA,NY
      GO TO 26
61    NY3=I-1
      NY=NY3
      GO TO 60
62    DETA=G*DETA
      DETAO2=DETA*0.5
      DO 63 I=1,88
      PBXY(3,I)=PBXY(2,I)
      FNXY(3,I)=FNXY(2,I)
      DSYXY(3,I)=DSYXY(2,I)
      DSNXY(3,I)=DSNXY(2,I)
      STRXY(3,I)=STRXY(2,I)
      PBXY(2,I)=PBXY(1,I)
      FNXY(2,I)=FNXY(1,I)
      DSYXY(2,I)=DSYXY(1,I)
      DSNXY(2,I)=DSNXY(1,I)
      STRXY(2,I)=STRXY(1,I)
      PBXY(1,I)=PB(I)
      FNXY(1,I)=FNS(I)
      DSYXY(1,I)=DSTRY(I)
      DSNXY(1,I)=DSTRN(I)
63    STRXY(1,I)=STRM(I)
      ETA3=ETA2
      ETA2=ETA1
      ETA1=ETA
      NY3=NY-3
      IF (STRM(NY3)) 65,65,64
64    IF (NETA-10) 27,65,65
65    RETURN
C
C
C
66    FORMAT (8F8.4,5E11.4)
67    FORMAT (17H1FIELD POINT DATA)
68    FORMAT (10F6.3,8F7.3)
69    FORMAT (5F12.5)
```

```
70    FORMAT (5X12HETA STEP NO I3,5X4HETA=F10.7,5X12HNO OF XI S =I3)
71    FORMAT (31HOSTARTING POINT SHOCK WAVE DATA)
72    FORMAT (7H1    ETA/F7.4/18HOFIRST CHECK BLOCK)
73    FORMAT (7H1    ETA/F7.4/19HOSECOND CHECK BLOCK)
74    FORMAT (F10.6,9E12.5)
75    FORMAT (10H1D2STR N Y/(10F11.5))
76    FORMAT (5HODPDY/(10F11.5))
77    FORMAT (7H1    ETA/F7.4/18HOTHIRD CHECK BLOCK)
78    FORMAT (19HOFIELD POINT OUTPUT)
79    FORMAT (6H1DSTRY/(10F11.5))
80    FORMAT (7HOD2STRY/(10F11.5))
81    FORMAT (7H1    ETA/F7.4)
82    FORMAT (120HO              X              R     MACH NO      VELOCITY
     1 TOT PRES      PRESSURE   TEMPERATURE         DENSITY     DELTA     THETA
     2)
83    FORMAT (111HO              X              R     MACH NO      VELOCITY
     1 TOT PRES      PRESSURE   TEMPERATURE         DENSITY     DELTA)
84    FORMAT (1H1)
85    FORMAT (1X,1HY,10X,4HSTRM,8X,5HDSTRY,7X,6HD2STRY,6X,5HDSTRN)
86    FORMAT (1X,1HY,7X,3HSSQ,9X,2HF1,10X,2HF2,10X,2HF3,10X,2HF4,10X,2HF
     15,10X,6HDSORDS,6X,6HDSORDP,6X,6HDSORDD)
87    FORMAT (1X,1HY,4X,2HD1,4X,2HD2,4X,2HD3,4X,2HD4,4X,2HD5,4X,2HD6,4X,
     12HD7,4X,2HD8,4X,2HD9,4X,3HD10,4X,3HD11,4X,3HD12,4X,3HD13,4X,3HD14,
     24X,3HD15,4X,4HDPDN,3X,6HD2STRN)
88    FORMAT (1X,1HY,6X,4HSAV1,4X,4HSAV2,4X,5HDSTRN,3X,4HSTRM,4X,2HPR,6X
     1,3HSSQ,5X,2HF1,6X,2HF2,9X,2HF3,9X,2HF4,9X,3HFNS,8X,3HDEN)
89    FORMAT (5X10HPARAMETERS,10X2HB=E18.6/25X5HDETA=E15.6/25X3HDY=E17.6
     1/25X2HG=E18.6)
      END
```

```
      OVERLAY (BLUNT,2,0)
      PROGRAM IVLS
      COMMON /BETA/ G,DETA,DY,RBODY,BBODY,DELTA,B,VN,GAMA,CM1,CM2,CM3,
     1CPR(88),ONE05,MDM
      DIMENSION X(100,10),Y(100,10),DF(100,10),VM(100,10),P(100,10),SR(
     1100,10),DUMMY(100,10,6),L(10),XA(10),YA(10),DFA(10),VMA(10),PA(10)
     2,SRA(10),DUMMZ(10,6),TH(100)
      DIMENSION ETA(10)
      COMMON /ALPHA/ DUMMY,DUMMZ,TH
      EQUIVALENCE (X(1,1),DUMMY(1,1,1)), (Y(1,1),DUMMY(1,1,2)), (DF(1,1)
     1,DUMMY(1,1,3)), (VM(1,1),DUMMY(1,1,4)), (P(1,1),DUMMY(1,1,5)), (SR
     2(1,1),DUMMY(1,1,6)), (XA(1),DUMMZ(1,1)), (YA(1),DUMMZ(1,2)), (DFA(
     31),DUMMZ(1,3)), (VMA(1),DUMMZ(1,4)), (PA(1),DUMMZ(1,5)), (SRA(1),
     4DUMMZ(1,6))
      RAD=57.29578
      REWIND 2
      ISLOPE=0
      SBODY=1.0/RBODY
      RBODY=1.0
      DELTA=DELTA*SBODY
      LBACK=0
      I=1
      J=1
    1 READ (2) X(I,J),Y(I,J),VM(I,J),SR(I,J),P(I,J),DF(I,J)
      X(I,J)=X(I,J)*SBODY
      Y(I,J)=Y(I,J)*SBODY
      IF (I-1) 3,3,2
    2 IF (Y(I,J)-Y(I-1,J)) 7,7,3
    3 IF (J-1) 4,4,5
    4 READ (2) TH(I)
    5 I=I+1
      IF (I-100) 1,1,6
    6 I=1
      L(J)=100
      GO TO 10
    7 IF (I-2) 12,12,8
    8 L(J)=I-1
      DO 9 K=1,6
      DUMMY(I,J+1,K)=DUMMY(I,J,K)
    9 DUMMY(I,J,K)=0.0
```

```
      I=2
10    J=J+1
      IF (J-10) 1,1,11
11    J=10
      GO TO 13
12    J=J-1
13    IMAX=0
      WRITE (6,54) VN,GAMA,RBODY,BBODY,DELTA
      CALL BBOUT (J,L)
      M=1
      N=J
14    LL=L(N)
      DO 15 I=1,LL
      IF (VM(I,N)-1.0) 15,15,16
15    CONTINUE
      WRITE (6,53) M,N,LL
16    CALL ENTRPG (VM(I-2,N),VM(I-1,N),VM(I,N),X(I-2,N),X(I-1,N),X(I,N),
     11.0,X1)
      CALL ENTRPG (VM(I-2,N),VM(I-1,N),VM(I,N),Y(I-2,N),Y(I-1,N),Y(I,N),
     11.0,Y1)
      IF (N-J) 18,17,17
17    X2=X1
      Y2=Y1
      SLOPET=1.0
      XMAX=X2
      YMAX=Y2
      ISB=I
      GO TO 20
18    SLOPES=((Y1-Y2)/(X1-X2))
      TLOPES=(ATAN(SLOPES))*RAD+175.0
      SLOPES=SIN(TLOPES/RAD)/COS(TLOPES/RAD)
      IF (SLOPES-SLOPET) 19,20,20
19    SLOPET=SLOPES
20    IF (I-IMAX) 22,21,21
21    IMAX=I
      XMAX=X1
      YMAX=Y1
22    N=N-1
      IF (N) 23,23,14
23    ETA(1)=1.0
```

```
      DO 24 I=2,J
      ETA(I)=ETA(I-1)-DETA
   24 DETA=DETA*G
      SLOPES=SLOPET
      XO=RBODY+DELTA
      WRITE (6,59)
      SLOPE=YMAX/(XMAX-XO)
      IF (SLOPE.GT.SLOPES) ISLOPE=1
   25 CALL ENTRPG (VM(ISB-1,J),VM(ISB-2,J),VM(ISB-3,J),X(ISB-1,J),X(ISB-
     12,J),X(ISB-3,J),ONE05,XQ)
      CALL ENTRPG (VM(ISB-1,J),VM(ISB-2,J),VM(ISB-3,J),Y(ISB-1,J),Y(ISB-
     12,J),Y(ISB-3,J),ONE05,YQ)
      SLOPE=YQ/(XQ-XO)
      IF (ISLOPE) 27,27,26
   26 XO=XQ-YQ/SLOPES
      SLOPE=SLOPES
   27 DO 39 I=1,J
      IF (I-J) 31,28,28
   28 XB=((SLOPE**2)*XO+RBODY+BBODY*DELTA)/(SLOPE**2+BBODY)
      XC=((SLOPE*XO)**2+(2.0*RBODY+BBODY*DELTA)*DELTA)/(SLOPE**2+BBODY)
      XB=XB-SQRT(XB**2-XC)
      LM=L(J)
      DO 29 L4=ISB,LM
      IF (XB-X(L4,J)) 30,30,29
   29 CONTINUE
      LBACK=LBACK+1
      K=LM+1-LBACK
      ONE05=FLOAT(IFIX(VM(K,J)*100.))/100.0
      WRITE (6,52) ONE05
      GO TO 25
   30 F3=L4
      L4=LM
      F2=L4-1
      F1=L4-2
      NZ=F1
      CALL ENTRPG (X(L4-2,J),X(L4-1,J),X(L4,J),F1,F2,F3,XB,FB)
      CALL DETER (L4-2,J,FB)
      GO TO 39
   31 RM=SLOPE/ETA(I)
      RP=RM*XO/SBODY
```

```
      RK=(1.0-B+B*(ETA(I)**2))
      TERM=B+RK*(RM**2)
      T1=B*(1.0+RK*RM*RP)/(B*TERM)
      T2=SQRT(B*RK*(TERM-(B*RP-RM)**2))/(B*TERM)
      XP=T1+T2
      XM=T1-T2
      IF (XP/XM) 32,35,35
   32 IF (XP) 33,33,34
   33 XI=XM
      GO TO 36
   34 XI=XP
      GO TO 36
   35 XI=AMIN1(XP,XM)
   36 ZI=SQRT((1.-((1.-B*XI)**2/RK))/B)
      Z=(ZI/DY)+0.5
      NZ=Z-1.0
      IF (NZ-L(I)+1) 37,38,38
   37 CALL DETER (NZ,I,Z)
      GO TO 39
   38 WRITE (6,58) NZ,L(I),I
      GO TO 51
   39 CONTINUE
   40 IF (NZ+2-L(J)) 42,42,41
   41 NZ=NZ-1
      GO TO 40
   42 SLOPE=(ATAN((RBODY-(XA(J)-DELTA)*BBODY)/YA(J)))*57.29578
      DSLOPE=(DFA(J)-SLOPE)/SQRT((XA(1)-XA(J))**2+(YA(1)-YA(J))**2)
      DO 43 I=2,J
   43 DFA(I)=DFA(I)-DSLOPE*SQRT((XA(I)-XA(1))**2+(YA(I)-YA(1))**2)
      IF (ISLOPE) 46,46,44
   44 DO 45 KJ=1,6
   45 DUMMZ(J,KJ)=DUMMY(ISB-1,J,KJ)
   46 CALL ENTRPG (Y(ISB-1,J),Y(ISB,J),Y(ISB+1,J),CPR(ISB-1),CPR(ISB),
     1CPR(ISB+1),YA(J),CD)
      WRITE (6,60)
      WRITE (6,55) VN,GAMA
      WRITE (6,61)
      WRITE (6,55) DELTA,RBODY,BBODY
      WRITE (6,57) CD,Y(ISB+1,J),CPR(ISB+1),Y(ISB,J),CPR(ISB),Y(ISB-1,J)
     1,CPR(ISB-1),YA(J)
```

```
      WRITE (6,62)
      DO 47 I=1,J
      WRITE (6,56) DUMMZ(I,1),DUMMZ(I,2),DUMMZ(I,3),DUMMZ(I,4),DUMMZ(I,6
     1)
47    CONTINUE
      A=0.0
      S=SR(1,J)
      LL=L(1)
      DO 48 NZ=1,LL
      IF (Y(NZ,1)-YA(1)) 48,49,49
48    CONTINUE
49    DUMMY(NZ,1,2)=DUMMZ(1,2)
      DUMMY(NZ,1,6)=DUMMZ(1,6)
      WRITE (6,63)
      WRITE (6,55) A,S,(DUMMY(I,1,2),DUMMY(I,1,6),I=1,NZ)
      NNZ=NZ+1
      WRITE (3) CM1,CM2
      K=J
      WRITE (3) K,NNZ,CD,VN,GAMA,DELTA,CM3
      WRITE (3) A,S,(DUMMY(I,1,2),DUMMY(I,1,6),I=1,NZ)
      DO 50 LLL=1,J
      I=J+1-LLL
      WRITE (3) DUMMZ(I,1),DUMMZ(I,2),DUMMZ(I,3),DUMMZ(I,4),DUMMZ(I,6)
50    CONTINUE
51    REWIND 3
C
C
52    FORMAT (37H0_INTERPOLATION MACH NO. DECREASED TO,G11.4,34HDUE TO L
     1IMITATION ON BODY MACH NO./)
53    FORMAT (5X32HDID NOT FIND SUPERSONIC POINT,M=I2/30X2HN=I2/29X3HLL=
     1I2)
54    FORMAT (1H15X39HCOMPLETE FLOW FIELD NORMALIZED BY RBODY///20X2HM=
     1F15.6/20X5HGAMA=F12.6/20X3HRB=E18.6/20X2HB=E19.6/20X6HDELTA=E15.6)
55    FORMAT (6E20.7)
56    FORMAT (5E20.7)
57    FORMAT (///5X3HCD=E15.6,5X28HBASED ON THE FOLLOWING TABLE//35X4HY(
     1I),11X6HCPR(I)/E45.6,E15.6/E45.6,E15.6/E45.6,E15.6//20X19HINTERPOL
     2ATED FOR Y=E20.6)
58    FORMAT (5X3HNZ=I5,5X5HL(I)=I5,2HI=I5/5X14HCANNOT PROCEED)
59    FORMAT (1H1)
```

```
60    FORMAT (1X,23HINITIAL VALUE LINE DATA/5X8HMACH NO.,12X5HGAMMA)
61    FORMAT (///5X5HDELTA,15X5HRBODY,15X5HBBODY)
62    FORMAT (///5X26HIVL DATA,STARTING AT SHOCK/9X1HX,19X1HY,19X5HDELTA
     1,15X8HMACH NO.,12X6HPT/PTO)
63    FORMAT (///1X18HMASS-ENTROPY TABLE)
      END
```

```
      SUBROUTINE DETER (N1,J,Z)
      DIMENSION DUMMY(100,10,6),DUMMZ(10,6),TH(100)
      COMMON /ALPHA/ DUMMY,DUMMZ,TH
      F1=N1
      F2=N1+1
      F3=N1+2
      N2=N1+1
      N3=N1+2
      DO 1 I=1,6
      CALL ENTRPG (F1,F2,F3,DUMMY(N1,J,I),DUMMY(N2,J,I),DUMMY(N3,J,I),Z,
     1DUMMZ(J,I))
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BBOUT (J,L)
      DIMENSION X(100,10),Y(100,10),DF(100,10),VM(100,10),P(100,10),SR(
     1100,10),DUMMY(100,10,6),L(10),XA(10),YA(10),DFA(10),VMA(10),PA(10)
     2,SRA(10),DUMMZ(10,6),TH(100)
      DIMENSION T(9)
      COMMON /ALPHA/ DUMMY,DUMMZ,TH
      COMMON /BETA/ G,DETB,DY,ARBE,BEE,CARD,AC,ACHM,GAMA,ALFA,ELAM,RINGS
     1,CPR(88),ONE05,MDM
      EQUIVALENCE (X(1,1),DUMMY(1,1,1)), (Y(1,1),DUMMY(1,1,2)), (DF(1,1)
     1,DUMMY(1,1,3)), (VM(1,1),DUMMY(1,1,4)), (P(1,1),DUMMY(1,1,5)), (SR
     2(1,1),DUMMY(1,1,6)), (XA(1),DUMMZ(1,1)), (YA(1),DUMMZ(1,2)), (DFA(
     31),DUMMZ(1,3)), (VMA(1),DUMMZ(1,4)), (PA(1),DUMMZ(1,5)), (SRA(1),
     4DUMMZ(1,6))
      GG6=GAMA-1.0
      GG2=GG6*0.5
      GG7=1.0/GAMA
      GG9=-GG6*GG7
      VMM=1.0+GG2*(ACHM**2)
      LIM=46
      J=J
      L(1)=L(1)
      DO 8 I=1,J
      MM=1
      IF (I-1) 1,1,2
    1 WRITE (6,30)
      GO TO 3
    2 WRITE (6,31)
    3 MAX=L(I)
      WRITE (6,28)
      DO 7 M=1,MAX
      IF (I.EQ.1) WRITE (6,27) TH(M)
      IF (I.GT.1) WRITE (6,28)
      TEM=VMM/(1.0+GG2*(VM(M,I)**2))
      VEL=SQRT(TEM*(VM(M,I)/ACHM)**2)
      DEN=(P(M,I)**GG7)/(SR(M,I)**GG9)
      WRITE (6,33) X(M,I),Y(M,I),VM(M,I),VEL,SR(M,I),P(M,I),TEM,DEN,DF(M
     1,I)
      MM=MM+1
      IF (MM-LIM) 7,7,4
    4 LIM=55
```

```
      MM=1
      WRITE (6,26)
      IF (I-1) 5,5,6
5     WRITE (6,30)
      GO TO 7
6     WRITE (6,31)
7     CONTINUE
8     WRITE (6,26)
      DO 24 IP=1,2
      GO TO (10,9), IP
9     WRITE (6,29)
      GO TO 11
10    WRITE (6,25)
11    DO 24 I=1,J
      MAX=L(I)
      GO TO (12,16), IP
12    DO 13 M=1,MAX
      IF (DUMMY(M,I,4)-1.0) 13,14,14
13    CONTINUE
      C3=MAX
      M3=MAX
      WRITE (6,32)
      GO TO 15
14    C3=M
      M3=M
15    M2=M3-1
      C2=M2
      M1=M2-1
      C1=M1
      CALL ENTRPG (DUMMY(M1,I,4),DUMMY(M2,I,4),DUMMY(M3,I,4),C1,C2,C3,
     11.0,AD)
      GO TO 17
16    AD=0
      C1=0.5
      C2=1.5
      C3=2.5
      M1=1
      M2=2
      M3=3
17    DO 18 IT=1,6
```

```
18    CALL ENTRPG (C1,C2,C3,DUMMY(M1,I,IT),DUMMY(M2,I,IT),DUMMY(M3,I,IT)
     1,AD,T(IT))
      IF (I-1) 19,19,23
19    GO TO (20,21), IP
20    CALL ENTRPG (C1,C2,C3,TH(M1),TH(M2),TH(M3),AD,TT)
      GO TO 22
21    TT=90.0
22    WRITE (6,30)
      IF (I.EQ.1) WRITE (6,27) TT
23    IF (I.GT.1) WRITE (6,28)
      T(8)=VMM/(1.0+GG2*T(4))
      T(7)=SQRT(T(8)*(T(4)/ACHM)**2)
      T(9)=(T(5)/T(6))**GG7
      WRITE (6,33) T(1),T(2),T(4),T(7),T(6),T(5),T(8),T(9),T(3)
24    CONTINUE
      RETURN
C
C
25    FORMAT (17HOSONIC POINT DATA)
26    FORMAT (1H1)
27    FORMAT (1H ,112XF9.5)
28    FORMAT ()
29    FORMAT (/////22HOSTAGNATION POINT DATA)
30    FORMAT (120H0            X              R    MACH NO     VELOCITY
     1 TOT PRES      PRESSURE  TEMPERATURE       DENSITY    DELTA    THETA
     2)
31    FORMAT (111H0            X              R    MACH NO     VELOCITY
     1 TOT PRES      PRESSURE  TEMPERATURE       DENSITY    DELTA)
32    FORMAT (1X,24HEXTRAPOLATED SONIC POINT)
33    FORMAT (1H+,2E13.6,0PF11.7,5E13.6,0P2F9.5)
      END
```

```
      OVERLAY (BLUNT,3,0)
      PROGRAM MOCS
C     EXECUTIVE PROGRAM FOR IDEAL GAS CHARACTERISTICS
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      COMMON /TRANS/ LO,L1,LL,MS
      COMMON /BETA/ D(7),ACHM,GAMA,ALFA,ELAMDA,RINGS,CPR(90),BDRAG
C     HOL(70)=RATIO OF PREVIOUS TO PRESENT MESH SIZE=HOL(I+2)/HOL(I+1)
C     HOL(69)=1,BODY INDICATE TO MESH P.,MESH SIZE FOR NEW SEG NOT FINAL
C     HOL(68)=SHOCK ANGLE FROM WAVE
C     HOL(63)=1 RADIAN
C     HOL(62)=PI
C     HOL(60)=NUMBER OF POINTS CALCULATED ON CURRENT CHARACTERISTIC.
C     HOL(59)-INDICATES POINT TO BE DROPPED AFTER SEC. SHOCK
C     HOL(58)=VMAX/V-FREE STREAM
C     HOL(57)=BODY SURFACE MASS FLOW RATIO
C     HOL(54)=INDICATOR TO CARRY OUT IVL TRANSITION
C     HOL(53)=STEP SIZE ALONG BODY DURING IVL TRANSITION
C     HOL(52)=BODY SURFACE TOTAL PRESSURE
C     HOL(50)=SUMMARY NO. FOR START OF CHECKOUT PRINT
C     HOL(49)=SUMMARY NO. FOR END OF CHECKOUT PRINT
C     HOL(48)=INDICATOR TO BODY SUBR TO CUT MESH SIZE DUE TO ENVELOPE SH
C     HOL(47)=INDICATOR TO BODY PT S-R THAT MESH SIZE IS NOT FINAL
C     HOL(46)=1,TELLS SHOCK PROGRAM TO TRY TERMINAL MODE
C     HOL(45)=+1 FOR EXTERNAL MODE, =-1 FOR INTERNAL MODE
C     HOL(40)=INDICATOR THAT A SPECIAL POINT FOLLOWS PREVIOUS ONE
C     HOL(39)=STEP SIZE ADJUSTMENT
C     HOL(3-28)=XSTARB OR MESH SIZE FOR SEG. 1,2,.....26 RESPECTIVELY
C     HOL(2)=1,INDICATE TO MAIN,SUBSONIC PT.,AVOID DOWN S. ANGULAR CALC.
C     HOL(1)=1,INDICATE TO BODY,SUBSONIC PT. ENCOUNTERED,NO (K-1,J-1)PT.
C     PART 1      INITIAL VALUE LINE
      JOLT=0
      XSTARB=0.0
      RAD=57.29578
```

```
      CALL SLITE (1)
      CALL PAGE (0)
      DO 1 I=1,72
    1 HOL(I)=0.0
      HOL(63)=1.5707963
      HOL(62)=3.1415927
      HOL(38)=0.5
      HOL(39)=0.5*(1.0+HOL(38))
      TEST=1.0E-06
      TESU=2.0E-06
      ISPEC=0
      DO 2 I=1,2
      DO 2 J=1,100
    2 ID(J,I)=0
      ILK=0
      ISEG=0
      KICK=0
      XSTAR=0.0
      TIN=0.0
      NB=0
      LOOP=0

      MEND=0
      REF=0
      IBT=0
      KPIO=1
      MID=1
      NSUM=1
      LO=1
      REF=1.0
      READ (5,74) INCASE,HOL(50),HOL(49),HOL(3)
      IPUNCH=4
      HOL(45)=+1.0
      DO 3 I=1,4
      AB(I)=0.0
      AC(I)=0.0
      AD(I)=0.0
    3 AE(I)=0.0
      READ (3) HOL(27),HOL(26)
      HOL(27)=HOL(27)/RAD
```

```
      HOL(26)=HOL(26)/RAD
      HOL(25)=HOL(27)
      HOL(24)=HOL(26)
      ALFA=HOL(27)
      ELAMDA=HOL(26)
      HOLDIF=HOL(27)-HOL(26)
      HOL(22)=1.0-SIN(ALFA+ELAMDA)
      HOL(27)=TAN(HOL(27))
      HOL(26)=SIN(HOLDIF)+1.0
      HOL(24)=HOL(26)-HOL(27)*COS(HOLDIF)
      HOL(23)=-HOLDIF
      CALL SHAPE
      NSEGM=NSEG-1
      EL=XB(NSEGM)
      XSPEC(1)=0.0
      XSPEC(1)=XB(1)
      XSPEC(2)=2.0
      READ (3) NIV,NSRV,CD,VMFS,GF(1),DET,RINGS
      RFIT=1.0
      AMUO=ANGLE(VMFS)
      GF(2)=GF(1)-1.0
      GF(3)=GF(1)+1.0
      GF(4)=(GF(1)-1.0)*0.5
      GF(5)=(GF(1)+1.0)*0.5
      GF(6)=(GF(1)-1.0)/(GF(1)+1.0)
      GF(7)=GF(1)*0.5
      GF(8)=(GF(1)+1.0)**2
      GF(9)=GF(1)*2.0
      GF(10)=GF(1)/(GF(1)-1.0)
      GF(11)=1.0/(GF(1)-1.0)
      GF(12)=2.0/(GF(1)-1.0)
      GF(13)=(GF(1)-1.0)/GF(1)
      GF(14)=SQRT((GF(1)-1.0)/2.0)
      GF(15)=1.0/GF(1)
      HOL(58)=1.0/VELOC(VMFS,GF(4))
      IF (HOL(50)) 4,5,5
    4 NSRV=-NSRV
      HOL(50)=ABS(HOL(50))
    5 MID=1
      WRITE (6,70)
```

```
      KPR=KPIO
      LEFT=0
      VFS=VMFS
      HOL(51)=LEFT
      PFS=(1.0+GF(4)*VMFS**2)**(-GF(10))
      RV2=GF(7)*(VMFS**2)*((1.0+GF(4)*(VMFS**2)))**(-GF(10))
      NEX=0
      CALL OUTPUT (1,0,0)
      REF2=REF**(MID+1)
      CD=CD*(R/REF)**(MID+1)
      DEM=MID
      IF (INCASE.GT.2) CALL OUTPUT (2,0,0)
C     DETERMINATION OF SONIC SHOCK PROPERTIES
      TSONIC=TSON(VMFS,GF(2),GF(3),GF(1))
      CALL WAVE (TSONIC,HOL(37),VMFS,PFS,1.0,1)
      PSONIC=PW
      VMSON=VMW
      DSONIC=DW
      BSONIC=ANGLE(VMW)
C     INTRODUCE CONSTANTS INTO ENTROP AND READ IN ENTROPY W/Y-REF
      CALL ENTROP (VELOC(VMFS,GF(4))*((R/RFIT)**(MID+1)),DEM,3,NSRV)
      NOW=1+NEX
      CALL INPUT (2,1,1)
      NS2=0
      ID(1,1)=0
      NNOW=1
      HOL(52)=SR(1,1)
      IF (X(1,1)-XB(1)) 6,7,7
6     X11=X(1,1)+HOL(41)
      DF(1,1)=ATAN((R-BL*X11)/SQRT((2.0*R-BL*X11)*X11))
7     CALL OUTPUT (6,1,1)
      IF (VM(1,1)-1.0) 8,8,9
8     NSKIP=1
      HOL(1)=1.0
      HOL(2)=1.0
      GO TO 10
9     NSKIP=0
10    NB=NB+1
      CPBNB=(P(NNOW,1)-PFS)/RV2
      YLAST=Y(NNOW,1)
```

```
      XLAST=X(NNOW,1)
      CP1=CPBNB*Y(NNOW,1)
      HOL(60)=1-LEFT+NEX
      DO 32 J=2,NIV
      NOW=HOL(60)
      HOL(60)=1.0
      CALL INPUT (1,1,2)
11    IF (NIV-J) 12,12,13
12    CALL WAVE (DF(1,2),HOL(37),VMFS,PFS,1.0,2)
      THS(1,2)=HOL(68)
      VM(1,2)=VMW
      P(1,2)=PW
      SR(1,2)=SW
      ID(1,2)=+1
      NS2=1
13    CALL OUTPUT (6,1,2)
      IF (NOW) 19,19,14
14    IF (HOL(2)) 15,15,19
15    CALL DROP (1,2,NJUMP,IDROP)
      GO TO (16,30,30,30), NJUMP
16    CALL OPER8 (IRD)
      IF (KICK-1) 17,20,32
17    GO TO (19,18,32), IRD
18    NOW=HOL(60)
      GO TO 23
19    HOL(2)=0.0
      CALL BODI
      NAW=HOL(60)+1.0
      IF (X(NAW,2).LE.HOL(22)) LO=NSUM
      L1=NSUM
      IF (KICK-1) 22,20,32
20    IF (INCASE-4) 21,32,32
21    INCASE=4
      GO TO 11
22    HOL(60)=HOL(60)+1.0
      NOW=HOL(60)
23    NB=NB+1
      CPBNB=(P(NOW,2)-PFS)/RV2
      CP2=CPBNB*Y(NOW,2)
      CD=(CP1+CP2)*(Y(NOW,2)-YLAST)/REF2+CD
```

```
      CP1=CP2                                                           L 202
      CALL OUTPUT (8,1,2)                                               L 203
      IF (INCASE-1) 30,27,24                                            L 204
24    IF (J-NIV) 26,25,25                                               L 205
25    THP=THS(1,2)*RAD                                                  L 206
      CALL OUTPUT (4,1,2)                                               L 207
      GO TO 30                                                          L 208
26    CALL OUTPUT (5,1,2)                                               L 209
      GO TO 30                                                          L 210
27    IF (J-NIV) 29,28,28                                               L 211
28    WRITE (6,72)                                                      L 212
      CALL OUTPUT (7,1,2)                                               L 213
      THP=THS(1,2)*RAD                                                  L 214
      WRITE (6,73) THP                                                  L 215
      WRITE (6,69)                                                      LA215
29    CALL OUTPUT (7,NOW,2)                                             L 216
C     SET FIRST DIM TO SECOND DIM. ELEMENTS THEN ALL SECOND DIM. TO 0   L 217
30    CALL CHANGE (7,1)                                                 L 218
      CALL CHANGE (5,1)                                                 L 219
      DO 31 K=1,2                                                       L 220
      THS(K,1)=THS(K,2)                                                 L 221
31    THS(K,2)=0.0                                                      L 222
      XLAST=X(NOW,1)                                                    L 223
      YLAST=Y(NOW,1)                                                    L 224
      NS2=0                                                             L 225
32    CONTINUE                                                          L 226
      IF (KICK) 33,33,68                                                L 227
33    HOL(54)=1.0                                                       L 228
      HOL(69)=1.0                                                       L 229
      MS=NSUM-2                                                         L 230
C     PART 2      SHOCK-TO BODY CALCULATIONS                            L 231
      TIN=1.0                                                           L 232
      HOL(51)=0.0                                                       L 233
34    IF (HOL(40)) 36,36,35                                             L 234
35    ILK=HOL(40)                                                       L 235
36    L5=1                                                              L 236
      NS2=0                                                             L 237
      IF (HOL(28).NE.0.0) XSTAR=XSTAR*1.1                               L 238
      IF (TIN-1.0) 38,38,37                                             L 239
37    NOW=NOWH                                                          L 240
```

```
C     RESET ALL FIRST DIM. ELEMENTS TO SAVED VALUES BEG. W/ELEMENT TWO
      CALL CHANGE (2,1)
      GO TO 39
38    NOWH=NOW
39    CALL SHOCK (1,L5)
      XLIM=HOL(24)+Y(1,2)*HOL(27)
      IGO=3
      IF ((X(1,2).GT.XLIM).OR.(NOW.LT.3)) GO TO 44
      IF (KICK-1) 42,40,68
40    IF (INCASE-4) 41,68,68
41    INCASE=4
      GO TO 36
42    IF (MEND) 46,46,43
43    IF (NOW-2) 45,45,46
44    CONTINUE
45    NOW=1
      GO TO 62
46    CALL OPER8 (IRD)
      IF (HOL(28)) 48,48,47
47    IF (HOL(60).EQ.2.0) GO TO 67
      GO TO 55
48    CONTINUE
      IF (KICK-1) 51,49,68
49    IF (JOLT) 50,50,68
50    KICK=0
      JOLT=1
      GO TO 40
51    GO TO (52,55,68,39), IRD
52    IF (MEND) 54,54,53
53    NOW=HOL(60)
      GO TO 62
54    CALL BODI
      NEW=HOL(60)+1.0
      IF (X(NEW,2).LE.HOL(22)) LO=NSUM
      IF (KICK-1) 56,49,68
55    NEW=HOL(60)
56    IF (INCASE-4) 58,57,58
57    WRITE (6,69)
      CALL OUTPUT (7,NEW,2)
58    NOOW=NOW
```

```
      NOW=NEW                                                              L 281
      IF (HOL(28)) 59,59,62                                                L 282
   59 CONTINUE                                                             L 283
      CALL MESH (IGO)                                                      L 284
      NOW=NOOW                                                             L 285
      ILK=ILK                                                              L 286
      GO TO (60,68,61,61), IGO                                             L 287
C     DROP ALL TEMPORARY SR ENTRIES                                        L 288
   60 CALL ENTROP (0.0,0.0,6,0)                                            L 289
      GO TO 34                                                             L 290
   61 NB=NB+1                                                              L 291
      NOW=NEW                                                              L 292
      L1=NSUM                                                              L 293
      CPBNB=(P(NOW,2)-PFS)/RV2                                             L 294
      CP2=CPBNB*Y(NOW,2)                                                   L 295
      CD=(CP1+CP2)*(Y(NOW,2)-YLAST)/REF2+CD                                L 296
      CP1=CP2                                                              L 297
C     RETAIN ALL TEMPORARY SR ENTRIES                                      L 298
   62 CALL ENTROP (0.0,0.0,5,0)                                            L 299
      XLAST=X(NOW,2)                                                       L 300
      YLAST=Y(NOW,2)                                                       L 301
C     SET FIRST DIM TO SECOND DIM. ELEMENTS THEN ALL SECOND DIM TO 0       L 302
      CALL CHANGE (7,1)                                                    L 303
      CALL CHANGE (5,1)                                                    L 304
      DO 63 K=1,2                                                          L 305
      THS(K,1)=THS(K,2)                                                    L 306
   63 THS(K,2)=0.0                                                         L 307
      THP=THS(1,1)*RAD                                                     L 308
      TIN=1.0                                                              L 309
      LOOP=0                                                               L 310
      CALL OUTPUT(8,1,1)                                                   L 3105
      IF (INCASE-1) 66,64,65                                               L 311
   64 WRITE (6,72)                                                         L 312
      CALL OUTPUT (7,1,1)                                                  L 313
      IF (NOW.EQ.1) GO TO 68                                               L 3135
      WRITE (6,71)                                                         L 314
      WRITE (6,69)                                                         L 315
      CALL OUTPUT (7,NOW,1)                                                L 316
      GO TO 66                                                             L 317
   65 CALL OUTPUT (4,1,1)                                                  L 318
```

```
66    CONTINUE
      IF (NOW-1) 68,68,34
67    MEND=1
      HOL(46)=1.0
      IGO=3
      GO TO 34
68    LL=NSUM-1
      BDRAG=CD
C
69    FORMAT (1X,10HBODY POINT)
70    FORMAT (1X,44(1H*),27HROTATIONALLY SYMMETRIC FLOW,56(1H*))
71    FORMAT (1H-)
72    FORMAT (1X,11HSHOCK POINT)
73    FORMAT (14X,12HTHETA SHOCK=E13.6)
74    FORMAT (I1,F3.0,F2.0,E18.6)
      END
```

```
      SUBROUTINE BODI
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
C     BODY  POINT  DETERMINATION
      IUP=0
      IDW=0
      IPDT=0
      IMM=0
      K=HOL(60)
      J=2
      G1=GF(1)
      IM=0
      X1=X(K,J)
      Y1=Y(K,J)
      IF (Y1.EQ.0.0) GO TO 1
      P1=P(K,J)
      AMU1=ANGLE(VM(K,J))
      D1=DF(K,J)
      VM1=VM(K,J)
      S1=SR(K,J)
      RAY1=(((P1/PFS)**GF(15))*(S1**GF(13))*GF(14)/SQRT(1.0+GF(4)*VM1**2
     1))*(Y1**MID)
1     IF (HOL(72)) 2,2,6
2     IF (HOL(1)) 4,4,3
3     L=1
      M=1
      HOL(1)=0.0
      GO TO 5
4     L=NOW
      M=J-1
5     S3=HOL(52)
6     IF (INCASE-4) 8,7,7
7     WRITE (6,124)
      WRITE (6,137) HOL(72),HOL(1),TIN,X1,Y1,P1,AMU1,D1,S3,K,J
```

```
8       IF (ILK) 10,10,9
9       I=ILK
        XTRY=XSPEC(I)
        GO TO 29
10      XTRY=XLAST+XSTARB*HOL(39)
        IF (XTRY-HOL(31)) 11,12,12
11      XTRY=HOL(31)
C       CONIC,  CUBIC,  ORRADIUS  TYPE  OF  BODY
12      DO 13 KK=1,NSEG
        I=KK
        IF (XTRY-XB(I)) 14,13,13
13      CONTINUE
        WRITE (6,123)
        GO TO 51
14      IF (XSTAR) 15,15,16
15      ISEG=I
        XSTAR=HOL(I+2)
        XSTARB=XSTAR
        GO TO 28
16      ISEGMI=ISEG-I
        IF (ISEGMI) 17,20,20
17      IMO=I-1
        DO 18 M=ISEG,IMO
        IF (XSPEC(M)) 19,18,19
18      CONTINUE
        M=I
19      XSTARB=HOL(M+2)
        ISEG=M
20      IF (HOL(54)) 24,24,21
21      HOL(54)=HOL(54)+1.0
        IF ((XSTARB/HOL(53))-2.0) 22,23,23
22      HOL(54)=0.0
        HOL(69)=0.0
        GO TO 34
23      XSTAR=XSTAR*2.0
        GO TO 34
24      IF (ISEGMI) 25,29,29
25      HOL(70)=HOL(I+2)/HOL(I+1)
        IF (HOL(70)-2.0) 26,27,27
26      XSTAR=XSTAR*HOL(70)
```

```
      HOL(69)=0.0
      GO TO 28
27    XSTAR=2.0*XSTAR
      HOL(69)=1.0
      HOL(70)=HOL(70)/2.0
28    IF (HOL(72)) 34,34,33
29    IF (HOL(69)) 34,34,30
30    IF (HOL(70)-1.0) 31,32,32
31    HOL(69)=0.0
      GO TO 28
32    XSTAR=2.0*XSTAR
      HOL(70)=HOL(70)/2.0
      GO TO 28
33    HOL(72)=0.0
      GO TO 122
34    N11=1
      X3=XTRY
      D3=D1
      VM3=VM1
      AMU3=AMU1
      P3=((1.0+GF(4)*(VM3**2))**(-GF(10)))*S3
      D1MU1=D1-AMU1
35    N1=1
36    J1SEG=IABS(J1(I))+1
      IF (I-1) 37,37,38
37    J1SEG=1
38    XZ=X3
      TANAV=TAN((D1MU1+D3-HOL(45)*AMU3)*.5)
      GO TO (39,46), J1SEG
39    BB=2.0*TANAV*(Y1-TANAV*X1)-2.0*R+2.0*BL*HOL(41)
      AAA=BL+TANAV**2
      CC=(Y1-TANAV*X1)**2+HOL(41)*(BL*HOL(41)-2.0*R)
      IF (AAA.NE.0.0) GO TO 40
      X3=-CC/BB
      GO TO 43
40    SQ=BB/(2.0*AAA)
      TQ=SQ**2-CC/AAA
      IF (TQ) 41,41,42
41    I=I+1
      GO TO 17
```

```
42    X3=-SQ-SQRT(TQ)
43    Y3=SQRT((2.0*R-BL*(X3+HOL(41)))*(X3+HOL(41)))
      IF (INCASE-4) 45,44,44
44    WRITE (6,130) X3,AAA,BB,CC
45    D3=ATAN((R-BL*(X3+HOL(41)))/Y3)
      GO TO 49
C     CUBIC  TYPE  BODY
46    IF (HOL(45)) 47,48,48
47    X3=(Y1-Y(L,M)+X(L,M)*AD(1)-X1*TANAV)/(AD(1)-TANAV)
      XX=X3-X(L,M)
      Y3=AD(1)*XX+AE(1)
      GO TO 77
48    XX=X3-XB(I-1)
      Y3=((AB(I)*XX+AC(I))*XX+AD(I))*XX+AE(I)
      D3=(3.0*AB(I)*XX+2.0*AC(I))*XX+AD(I)
      X3=X3-(Y3-(X3-X1)*TANAV-Y1)/(D3-TANAV)
      D3=ATAN(D3)
49    IF (N1-99) 52,50,50
50    WRITE (6,131) N1
51    KICK=KICK+1
      GO TO 122
52    IF (INCASE-4) 54,53,53
53    WRITE (6,132) N1,X3,I
54    N1=N1+1
      IF (ABS(X3)-TEST) 55,55,56
55    IF (ABS(X3-XZ)-TEST) 57,38,38
56    IF (ABS((XZ/X3)-1.0)-TEST) 57,38,38
57    IF (LOOP+IPDT) 58,58,69
58    IF (X3-XB(I)) 60,60,59
59    IUP=I
      I=I+1
      GO TO 63
60    IF (I-1) 69,69,61
61    IF (XB(I-1)-X3) 69,62,62
62    IDW=I
      I=I-1
63    IF (INCASE-4) 65,64,64
64    WRITE (6,133) I
65    IPDT=IUP*IDW
      IF (IPDT) 66,36,66
```

```
66    IF (ILK) 67,67,68
67    I=IUP
      GO TO 36
68    I=ILK
      GO TO 36
69    IF (N11-1) 70,70,76
70    IF (VM(NOW,1)-1.0) 71,71,74
71    P3=P(NOW,1)+(S3-PFS)*SIN(D3+DF(NOW,1))*SIN(D3-DF(NOW,1))
72    VM3=SQRT(GF(12)*((P3/S3)**(-GF(13))-1.0))
      AMU3=ANGLE(VM3)
      IF (INCASE-4) 76,73,73
73    WRITE (6,141) P3,D3,VM3,AMU3
      GO TO 76
74    DNU=DF(NOW,1)-D3
      T1=VM(NOW,1)**2
      P3=(1.0-((GF(1)*T1)/SQRT(T1-1.0))*DNU)*P(NOW,1)
      IF (P3) 75,75,72
75    P3=P(NOW,1)
      GO TO 72
76    D3M1=D3-D1
77    DS=SQRT((X3-X1)**2+(Y3-Y1)**2)
      YAV=(Y1+Y3)*0.5
      AVD=(D1+D3)*0.5
      SAVD=SIN(AVD)
      IF (HOL(45)) 78,79,79
78    B2=SAVD*SIMU*DS/YAV
      RT=B1+B2
      B3=D1-RT
      RESID=D3-B3
      GO TO 80
79    AVMU=(AMU1+AMU3)
      S2MU=SIN(AVMU)
      AVMU=0.5*AVMU
      SIMU=SIN(AVMU)
      B1=G1/S2MU
      B2=D3M1-DEM*SAVD*SIMU*DS/YAV
      RT=B1*B2
      RESID=RT-ALOG(P3/P1)
80    IF (ABS(RESID)-TESU) 105,81,81
81    IF (INCASE-4) 83,82,82
```

```
82    WRITE (6,136) RESID,P3,D3M1,DS,YAV,AVMU,AVD,SAVD,S2MU,SIMU,B1,B2,
     1RT,S3
83    IF (HOL(45)) 84,85,85
84    D3=B3
      GO TO 102
85    IF (N11-1) 86,86,89
86    R1=RESID
      P31=P3
      IF (ABS(RT)-88.028) 88,87,87
87    HOL(61)=1.0
      GO TO 122
88    P3=P1*EXP(RT)
      HOL(61)=0.0
      GO TO 93
89    IF (RESID/R1) 90,90,88
90    R2=RESID
      P32=P3
      P3=P31-R1*(P3-P31)/(R2-R1)
      IF (INCASE-4) 92,91,91
91    WRITE (6,127) P3,P32,R2,P31,R1
92    P31=P32
      R1=R2
      IM=0
      IMM=0
93    VM3=VMACH(P3,S3,GF(12),GF(13))
      IF (VM3-1.0) 95,95,94
94    IM=0
      GO TO 99
95    IM=IM+1
      IF (TIN) 97,97,96
96    HOL(61)=1.0
      GO TO 122
97    VM3=1.0
      AMU3=1.5707963
      P3=((1.0+GF(4))**(-GF(10)))*S3
      IF (IM-25) 100,100,98
98    WRITE (6,140)
      GO TO 51
99    AMU3=ANGLE(VM3)
100   IF (INCASE-4) 102,101,101
```

```
101   WRITE (6,128) N11
      WRITE (6,129) X3,Y3,D3,P3,RESID,AMU3
102   N11=N11+1
      IF (N11-50) 103,103,104
103   IF (HOL(45)) 38,35,35
104   WRITE (6,134) RESID,RT,I
      GO TO 51
105   IF (HOL(45)) 116,106,106
106   IF (IMM+IM) 108,108,107
107   WRITE (6,135) IMM,IM,VM3
      GO TO 51
108   IF (ILK) 109,109,116
109   IF (X3-XB(I)) 112,116,110
110   I=I+1
      IF (I-NSEG) 35,35,111
111   WRITE (6,126) X3,XB(I-1)
      GO TO 51
112   IF (I-2) 113,114,114
113   GO TO 116
114   IF (X3-XB(I-1)) 115,116,116
115   I=I-1
      GO TO 35
116   IF (INCASE-2) 118,118,117
117   WRITE (6,125)
      WRITE (6,129) X3,Y3,D3,P3,RESID,VM3
118   X(K+1,J)=X3
      Y(K+1,J)=Y3
      DF(K+1,J)=D3
      P(K+1,J)=P3
      VM(K+1,J)=VM3
      SR(K+1,J)=S3
      ID(K+1,J)=0
      HOL(53)=X3-X(NOW,1)
      IF (INCASE-4) 120,119,119
119   WRITE (6,139) XSTARB,HOL(53),HOL(54),ISEGMI
120   CONTINUE
      RAY2=(((P3/PFS)**GF(15))*(S3**GF(13))*GF(14)/SQRT(1.0+GF(4)*VM3**2
     1))*(Y3**MID)
      DS=SQRT((X3-X1)**2+(Y3-Y1)**2)
      FMASS=VMASS(K)-(1.0+DEM)*0.5*(RAY1+RAY2)*DS
```

```
      VMASS(K+1)=FMASS
      HOL(57)=FMASS/VMASS(1)
      IF (INCASE-4) 122,121,121
121   WRITE (6,138) FMASS,VMASS(1),HOL(57)
122   RETURN
C
C
C
123   FORMAT (29H0 X1 LIES BEYOND LAST SEGMENT)
124   FORMAT (1X125(1H*))
125   FORMAT (1X,13H***BODY POINT)
126   FORMAT (30H POINT LIES BEYOND END OF BODY,3HX3=E15.7,3HXB=E15.7)
127   FORMAT (16H INTERPOLATED P=E15.6/4H P3=E15.7,4H R2=E15.6,5(1H ),5H
     1 P31=E15.6,4H R1=E15.6)
128   FORMAT (12H0 PITERATIONI5)
129   FORMAT (5H0 X3=E15.8,4H Y3=E15.8,4H D3=E15.8,4H P3=E15.8,6HRESID=
     1E15.8,6H AMU3=E15.8)
130   FORMAT (5(1H ),4H X3=E15.6,4H AA=E15.6,4H BB=E15.6,4H CC=E15.6)
131   FORMAT (19H0 ITERATION  FAILEDI5)
132   FORMAT (1H05X3HN1=I5/9X3HX3=E15.6,5X5HISEG=I5)
133   FORMAT (5X20HSEGMENT CHANGE,ISEG=I5)
134   FORMAT (61H0FAILURE IN BODY PROG. EXCEEDED 50 ITERATIONS IN OUTER
     1LOOP. 7H RESID=E13.6,5H  RT=E13.6,10H  SEGMENT=I3)
135   FORMAT (5H0IMM=I3,4H IM=I3,5H VM3=E13.6)
136   FORMAT (10H RESIDUAL=E13.6,6H    P3=E13.6,6H D3M1=E13.6,6H    DS=
     1E13.6,6H  YAV=E13.6,6H AVMU=E13.6/6X4HAVD=E13.6,6H SAVD=E13.6,6H S
     22MU=E13.6,6H SIMU=E13.6/7X3HB1=E13.6,6H    B2=E13.6,6H    RT=E13.6,6
     3H    S3=E13.6,24H IN BODY AT STATEMENT 35)
137   FORMAT (54H0BODY PROG. IF HOL(72)=1 FULL CHAR READ,ONLY 1ST LINE
     143HBELOW VALID.  IF HOL(1)=1 SUBS.PT.-HOL(72)=F3.1,8H HOL(1)=F4.1,
     25H TIN=F4.1/4H X1=E12.5,4H Y1=E12.5,4H P1=E12.5,6H AMU1=E12.5,4H D
     31=E12.5,/4H S3=E12.5,4H  K=I5,4H  J=I5)
138   FORMAT (22H RATIO=FMASS/VMASS(1)=E13.6,1H/E13.6,1H=E13.6)
139   FORMAT (8H XSTARB=E13.6,9H HOL(53)=E13.6,9H HOL(54)=E13.6,8H ISEGM
     1!=I5)
140   FORMAT (5X5(3H(*)),14HIM EXCEEDED 25)
141   FORMAT (5X16HFIRST APPROX P3=E15.6/21X3HD3=E15.6/20X4HVM3=E15.6/
     119X5HAMU3=E15.6)
      END
```

```
      SUBROUTINE OPER8 (IRD)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      LIMIT=0
      LEFT=HOL(51)
      IF (TIN) 1,1,2
    1 I1=1+LEFT
      GO TO 3
    2 I1=2
    3 DO 6 I=I1,NOW
      IF (HOL(59)) 5,5,4
    4 HOL(59)=HOL(59)-1.0
      GO TO 6
    5 L=1
      CALL FIELD (I,L)
      IH=HOL(60)
      XLIM=HOL(24)+Y(IH,2)*HOL(27)
      IF ((X(IH,2).GT.XLIM).AND.(I.LT.NOW)) HOL(28)=1.0
      IF ((X(IH,2).GT.XLIM).AND.(I.LT.NOW).AND.(TIN.GT.0.)) LIMIT=LIMIT+
     11
      IF (LIMIT.EQ.2) GO TO 7
    6 CONTINUE
    7 IRD=1
      IF (KICK.GT.0) IRD=3
      RETURN
      END
```

```
      SUBROUTINE FIELD (IB,JB)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      IBALL=0
      LBLOCK=0
      IND=0
      TEST=.000001
      TESU=2.0E-05
      J=JB
      I=IB
      L=HOL(60)
      IF (L-1) 1,1,3
    1 IF (TIN) 2,2,3
C     CALL ENTROP TO GET MASS AT SHOCK POINT
    2 CALL ENTROP (SR(1,J+1),VMASS(1),2,0)
    3 M=J+1
      PR=P(L,M)
      SZ=SR(L,M)
      WR=VM(L,M)
      XR=X(L,M)
      YR=Y(L,M)
      X1=X(L,M)
      Y1=Y(L,M)
      D1=DF(L,M)
      P1=P(L,M)
      S1=SR(L,M)
      W1=VM(L,M)
      K=HOL(60)
      X2=X(I,J)
      Y2=Y(I,J)
      D2=DF(I,J)
      P2=P(I,J)
      S2=SR(I,J)
      W2=VM(I,J)
```

```
      IT=0
      I1=0
      AMU1=ANGLE(W1)
      AMU2=ANGLE(W2)
      RAY1=(((PR/PFS)**GF(15))*(SZ**GF(13))*GF(14)/SQRT(1.0+GF(4)*WR**2)
     1)*(YR**MID)
      IF (INCASE-4) 5,4,4
4     WRITE (6,104) RAY1,VMASS(K),K
5     P3=0.5*(P1+P2)
      S3=(S1+S2)*0.5
      W3=VMACH(P3,S3,GF(12),GF(13))
      AMU3=ANGLE(W3)
      D12=0.5*(D1+D2)
      D3=D12
      P1L=ALOG(P1)
      P2L=ALOG(P2)
C     BEGIN GEOMETRIC PROCESS
6     IF (INCASE-4) 8,7,7
7     WRITE (6,101) IT
      WRITE (6,99) X3,Y3,P3,W3,D3,S3
8     A13=0.5*(AMU1+AMU3)
      A23=0.5*(AMU2+AMU3)
      SA23=SIN(A23)
      S2A23=SIN(2.0*A23)
      S2A13=SIN(2.0*A13)
      SA13=SIN(A13)
      D13=0.5*(D1+D3)
      S13=0.5*(D1-AMU1+D3-AMU3)
      S23=0.5*(D2+AMU2+D3+AMU3)
      D23=0.5*(D2+D3)
      SD23=SIN(D23)
      SD13=SIN(D13)
      TAN2=TAN(S23)
      TAN1=TAN(S13)
      DELS=(S2-S1)*0.1
      SBASIC=S1
      IFM=0
      IF (TAN2-2.0) 9,9,11
9     IF (TAN1+2.0) 10,10,13
C     NEITHER TANGENT GREATER THAN 2
```

```
10    X3=(Y2-Y1-(X2*TAN2-X1*TAN1))/(TAN1-TAN2)
      Y3=(TAN1*TAN2*(X1-X2)+Y2*TAN1-Y1*TAN2)/(TAN1-TAN2)
      GO TO 15
11    IF (TAN1+2.0) 12,12,14
C     BOTH TANGENTS GREATER THAN 2
12    X3=(X1*TAN1-X2*TAN2+Y2-Y1)/(TAN2-TAN1)
      Y3=((X2-X1)*TAN2*TAN1+Y1*TAN2-Y2*TAN1)/(TAN2-TAN1)
      GO TO 15
C     UPPER TANGENT ECEEDS 2
13    X3=((Y2-Y1)/TAN1+X1-X2*TAN2/TAN1)/(1.0-TAN2/TAN1)
      Y3=((X2-X1)*TAN2+Y1*TAN2/TAN1-Y2)/(TAN2/TAN1-1.0)
      GO TO 15
C     LOWER TANGENT EXCEEDS 2
14    X3=((Y2-Y1)/TAN2+X1*TAN1/TAN2-X2)/(TAN1/TAN2-1.0)
      Y3=((X2-X1)*TAN1+(Y1-Y2*TAN1/TAN2))/(1.0-TAN1/TAN2)
C     BEGIN COMPATABILITY EQUATIONS
15    IF (INCASE-4) 17,16,16
16    WRITE (6,97) X3,Y3,S3,W3
17    Y23=0.5*(Y2+Y3)
      Y13=0.5*(Y1+Y3)
      DOY23=SQRT((Y2-Y3)**2+(X2-X3)**2)/Y23
      DOY13=SQRT((Y1-Y3)**2+(X1-X3)**2)/Y13
      SXA=S3
      IF (LBLOCK) 18,18,32
18    T1=S2A23+S2A13
      T2=SD13*DOY13
      T3=SD23*DOY23
      T4=GF(15)*(P2L-P1L)
      T5=(D1*S2A23+D2*S2A13)
      T6=0.5*(S2A23*S2A13)
      D3=(T6*(T4+DEM*(T2/COS(A13)-T3/COS(A23)))+T5)/T1
      T7=S2A13*P1L+S2A23*P2L
      T8=D2-D1
      T9=T2*SA13
      T10=T3*SA23
      RT=(T7+GF(9)*(T8-DEM*(T9+T10)))/T1
      RESID=ALOG(P3)-RT
      IF (IT-1) 22,23,19
19    IF (ABS(RESID)-TEST) 20,20,21
20    IF (ABS(S3/S3I-1.0)-TEST) 78,78,21
```

```
21     IF (IT) 22,22,23
22     P31=P3
       P3=EXP(RT)
       R1=RESID
       GO TO 32
23     R2=RESID
       IF (R1) 24,26,24
24     IF (RESID/R1) 25,25,31
25     P32=P3
       DR=R2-R1
       IF (DR) 28,26,28
26     IF (ABS(R2)-TEST) 78,78,27
27     WRITE (6,98)
       GO TO 74
28     P3=P31-R1*(P3-P31)/DR
       IF (INCASE-4) 30,29,29
29     WRITE (6,91) P3,P32,R2,P31,R1
30     P31=P32
       R1=R2
       GO TO 32
31     P3=EXP(RT)
32     DS=SQRT((X3-XR)**2+(Y3-YR)**2)
       S3I=S3
       KSR=0
       KBLOCK=0
       IF (INCASE-4) 34,33,33
33     WRITE (6,96) T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,D3,RT
34     IF (IFM) 36,36,35
35     S3=SBASIC+DELS*FLOAT(IFM-1)
       SS3=S3
36     W3=VMACH(P3,S3,GF(12),GF(13))
       RAY2=(((P3/PFS)**GF(15))*(S3**GF(13))*GF(14)/SQRT(1.0+GF(4)*(W3**2
      1)))*(Y3**MID)
       FMASS=VMASS(K)-(1.0+DEM)*0.5*(RAY1+RAY2)*DS
       CALL ENTROP (S3,FMASS,1,0)
       KSR=KSR+1
       IF (INCASE-4) 38,37,37
37     WRITE (6,106) RAY2,FMASS,S3,DS
38     CONTINUE
       IF (KSR-1) 39,39,40
```

```
39    SIT=S3
      GO TO 34
40    IF (IFM-1) 52,42,41
41    SH=SG
      DELH=DELG
42    SG=SS3
      DELG=SG-S3
      IF (IFM-1) 50,50,43
43    IF (INCASE-4) 45,44,44
44    WRITE (6,92) IFM,SG,DELG,SH,DELH,S3
45    IF (DELG*DELH) 51,51,46
46    IF (ABS(DELH)-ABS(DELG)) 47,50,50
47    DELS=-DELS
      IBALL=IBALL+1
      IF (IBALL-1) 48,48,49
48    IFM=1
      SBASIC=SH
      GO TO 34
49    WRITE (6,105)
      KICK=1.0
      GO TO 86
50    IBALL=0
      IFM=IFM+1
      GO TO 34
51    S3=SG-DELG*((SG-SH)/(DELG-DELH))
      SIT=S3
      KBLOCK=KBLOCK+1
      DELS=(SG-SH)*0.1
      SBASIC=SH
      IFM=0
      GO TO 34
52    IF (ABS((S3/SIT)-1.0)-TEST) 59,59,53
53    IF (KSR-10) 39,39,54
54    IF (KBLOCK) 56,56,55
55    IF (KBLOCK-10) 56,56,58
56    IFM=1
      ITSAVE=IT
      TEST=TESU
      IT=2
      IF (LBLOCK) 57,57,34
```

```
57    LBLOCK=1
      GO TO 34
58    WRITE (6,93)
      GO TO 85
59    IF (INCASE-4) 61,60,60
60    WRITE (6,90) P3,D3,S3,R2,R1,RAY2,FMASS
61    IMM=0
      IF (LBLOCK-1) 70,62,63
62    SXB=SXA
      DXB=S3-SXA
      KBLOCK=0
      LBLOCK=2
      GO TO 70
63    KBLOCK=0
      LBLOCK=LBLOCK+1
      DXA=S3-SXA
      SAVG=0.5*(SXA+SXB)
      IF (ABS(DXA/SAVG)-TESU) 64,64,65
64    LBLOCK=0
      IT=ITSAVE
      GO TO 70
65    IF (LBLOCK-25) 67,67,66
66    WRITE (6,94)
      GO TO 85
67    S3=SXA-DXA*((SXA-SXB)/(DXA-DXB))
      W3=VMACH(P3,S3,GF(12),GF(13))
      IF (INCASE-4) 69,68,68
68    WRITE (6,95) SXA,SXB,DXA,DXB,S3
69    SXB=SXA
      DXB=DXA
70    IF (W3-1.0) 71,71,75
71    I1=I1+1
      IF (I1-10) 72,72,73
72    P3=((1.0+GF(4))**(-GF(10)))*S3
      W3=1.0
      AMU3=1.5707963
      D3=D12
      IMM=IMM+1
      GO TO 6
73    WRITE (6,102) W3,P3,S3
```

```
74    KICK=KICK+1
      GO TO 86
75    AMU3=ANGLE(W3)
      IT=IT+1
      I1=0
      IF (IT-50) 6,6,76
76    IF (IND-4) 77,88,88
77    IND=IND+1
      IT=1
      TEST=TEST*10.0
      GO TO 6
78    IF (IMM+I1) 79,79,87
79    HOL(60)=HOL(60)+1.0
      L=HOL(60)
      IF (HOL(45)) 80,81,81
80    M=J
      GO TO 82
81    M=J+1
82    X(L,M)=X3
      Y(L,M)=Y3
      P(L,M)=P3
      VM(L,M)=W3
      DF(L,M)=D3
      SR(L,M)=S3
      ID(L,M)=ID(IB,JB)
      VMASS(L)=FMASS
      IF (IND) 84,84,83
83    WRITE (6,103) IND,L
84    IF (INCASE-4) 86,85,85
85    WRITE (6,99) X3,Y3,P3,W3,D3,S3
86    RETURN
87    WRITE (6,100) IMM,P3
88    WRITE (6,89)
      GO TO 74
C
C
C
89    FORMAT (40H EXCEEDED 3*50 ITERATIONS IN FIELD POINT)
90    FORMAT (4H P3=E15.6,3HD3=E15.6,3HS3=E15.6,3HR2=E15.6,3HR1=E13.6,6H
     1 RAY2=E13.6,7H FMASS=E13.6)
```

```
91    FORMAT (16H INTERPOLATED P=E15.6/4H P3=E15.7,4H R2=E15.6,5(1H ),5H
     1 P31=E15.6,4H R1=E15.6)
92    FORMAT (5X15,(5E15.7))
93    FORMAT (1X20(1H+),37HFAILED TO CONVERGE S3 IN SPECIAL LOOP)
94    FORMAT (5X20(1H-),31HFAILED TO CONVERGE D3 SLOW LOOP)
95    FORMAT (1X10(1H-),4HSXA=E15.6,5X4HSXB=E15.6,5X4HDXA=E15.6,5X4HDXB=
     1E15.6,5X3HS3=E15.6)
96    FORMAT (6E20.6/6E20.6)
97    FORMAT (30X3HX3=E15.7,5X3HY3=E15.7,5X3HS3=E15.7,5X3HW3=E15.7)
98    FORMAT (20H DR FAILURE IN FIELD)
99    FORMAT (4H0X3=E15.8,3HY3=E15.8,3HP3=E15.8,3HM3=E15.8,3HD3=E15.8,3H
     1S3=E15.8)
100   FORMAT (5X20HFAILED IN FIELD,IMM=I5/4H P3=E15.6)
101   FORMAT (12H0 ITERATION I3)
102   FORMAT (55H TRIED 10 TIMES TO CONVERGE POINT BUT STILL SUBSONIC.
     13HW3=E13.6,7H     P3=E13.6,7H     S3=E13.6)
103   FORMAT (52H0****NOTE****CONVERGENCE CRITERIA MULTIPLIED BY 10**I1,
     110H FOR POINTI3)
104   FORMAT (6H RAY1=E12.6,10H VMASS(K)=E12.6,3H K=I3)
105   FORMAT (1X29HOSCILLATION IN FIELD FOR DELS)
106   FORMAT (7H RAY2 =E15.5,5X7HFMASS =E15.5,5X4HS3 =E15.5,5X3HDS=E15.6
     1)
      END
```

```
FUNCTION VMACH (P,SR,G12,G13)
VMACH=SQRT(G12*((P/SR)**(-G13)-1.0))
RETURN
END
```

```
      SUBROUTINE ENTROP (VAR1,VAR2,I1,I2)
C     *************************IDEAL GAS VERSION **************************
      DIMENSION QUAN(2,100),SSRG(100),SGMASS(100)
C     WHERE QUAN(1,I)=S/R,QUAN(2,I)=FMASS
      SRG=VAR1
      GMASS=VAR2
      I=I1
      GO TO (2,19,34,40,73,75), I
C     DETERMINE MAX MASS FOR INTERNAL FLOW OPTION
1     J=J+1
2     IF (GMASS) 3,4,4
3     SRG=QUAN(1,1)
      GO TO 72
4     CONTINUE
      IF (QUAN(2,J)-GMASS) 5,5,17
5     IF (J-NSRV) 1,6,6
6     IF (QUAN(2,J)-GMASS) 7,8,7
7     CALL PAGE (1)
      WRITE (6,78) GMASS,QUAN(1,J),QUAN(2,J)
8     SRG=QUAN(1,J)
      GO TO 72
9     J=J-1
      IF (QUAN(2,J)-GMASS) 10,8,17
10    IF (J+2-NSRV) 11,11,13
11    IF (QUAN(2,J+1)-QUAN(2,J+2)) 12,13,12
12    CALL ENTRPG (QUAN(2,J),QUAN(2,J+1),QUAN(2,J+2),QUAN(1,J),QUAN(1,J+
     11),QUAN(1,J+2),GMASS,SRG)
      GO TO 72
13    IF (J-1) 14,14,15
C     INTERPOLATE BETWEEN TWO VALUES-BEFORE AND AFTER MASS
14    SRG=(GMASS-QUAN(2,J))*(QUAN(1,J+1)-QUAN(1,J))/(QUAN(2,J+1)-QUAN(2,
     1J))+QUAN(1,J)
      GO TO 72
15    IF (QUAN(2,J-1)-QUAN(2,J)) 16,14,16
16    J=J-1
      GO TO 12
17    IF (J-1) 18,18,9
18    CALL PAGE (1)
      WRITE (6,79) GMASS,QUAN(1,J),QUAN(2,J)
      GO TO 8
```

```
C       DETERMINE GMASS CORRESPONDING W/CURRENT SRG
19      IF (QUAN(1,JJ)-SRG) 31,22,20
20      IF (JJ-1) 21,21,23
21      WRITE (6,85) SRG,QUAN(2,JJ),QUAN(1,JJ)
22      GMASS=QUAN(2,JJ)
        GO TO 72
23      JJ=JJ-1
        IF (QUAN(1,JJ)-SRG) 24,22,20
24      IF (JJ+2-NSRV) 25,25,27
25      IF (QUAN(2,JJ+1)-QUAN(2,JJ+2)) 26,27,26
26      CALL ENTRPG (QUAN(1,JJ),QUAN(1,JJ+1),QUAN(1,JJ+2),QUAN(2,JJ),QUAN(
       12,JJ+1),QUAN(2,JJ+2),SRG,GMASS)
        GO TO 72
27      IF (JJ-1) 30,30,28
28      IF (QUAN(1,J)-QUAN(1,J-1)) 29,30,29
29      JJ=JJ-1
        GO TO 26
30      GMASS=(SRG-QUAN(1,J))*(QUAN(2,J+1)-QUAN(2,J))/(QUAN(1,J+1)-QUAN(1,
       1J))+QUAN(2,J)
        GO TO 72
31      IF (JJ-NSRV) 32,33,33
32      JJ=JJ+1
        GO TO 19
33      CALL PAGE (1)
        WRITE (6,86) SRG,QUAN(2,JJ),QUAN(1,JJ)
        GMASS=QUAN(2,JJ)
        GO TO 72
C       READ IN ENTROPY W/CORRESPONDING R-REF VALUES
34      NSRV=IABS(I2)
        JJ=1
        J=1
        QUAN(2,1)=0.0
        QUAN(1,1)=0.0
        INCASE=I2
        MI=VAR2+1.0
        CON1=VAR1
        IF (NSRV-1) 35,35,36
35      NSRV=0
        I2=0
        GO TO 72
```

```
36    READ (3) (QUAN(2,K),QUAN(1,K),K=1,NSRV)
      I2=NSRV
      J=NSRV
      DO 37 K=1,NSRV
      IF (QUAN(1,1)) 38,37,37
37    QUAN(2,K)=QUAN(2,K)**MI*CONI
38    QUAN(1,1)=ABS(QUAN(1,1))
      WRITE (6,81) NSRV
C     SAVE CURRENT GMASS-S/R VALUES IN TABLE
      NSRVS=NSRV
      DO 39 L=1,NSRVS
      SSRG(L)=QUAN(1,L)
39    SGMASS(L)=QUAN(2,L)
      GO TO 67
C     ADD NEW VALUES OF S/R AND FMASS
40    DO 49 K=1,NSRV
      IF (QUAN(2,K)-GMASS) 48,41,53
41    KK=K
      L=K
      IF (QUAN(1,K)-SRG) 45,48,42
42    L=L-1
      IF (L-1) 53,53,43
43    IF (QUAN(2,L)-QUAN(2,L-1)) 42,44,42
C     POINTS TO BE DROPPED RANGE FROM L TO K-1 INCL.
44    QUAN(2,L)=GMASS
      QUAN(1,L)=SRG
      IK=L
      IF (NSRV-KK) 52,50,50
45    L=L-1
      IF (L-1) 48,48,46
46    IF (QUAN(2,L)-QUAN(2,L-1)) 45,47,45
47    QUAN(1,L)=QUAN(1,K)
      QUAN(2,L)=QUAN(2,K)
      L=L+1
      KK=K+1
      GO TO 44
48    IF (K-NSRV) 49,55,55
49    CONTINUE
      GO TO 56
50    DO 51 LL=KK,NSRV
```

```
      L=L+1
      QUAN(1,L)=QUAN(1,LL)
51    QUAN(2,L)=QUAN(2,LL)
      NSRV=L
      GO TO 56
52    NSRV=L+1
      GO TO 56
53    JL=-1
      DO 54 KK=K,NSRV
      JKL=NSRV-JL
      QUAN(1,JKL)=QUAN(1,JKL-1)
      QUAN(2,JKL)=QUAN(2,JKL-1)
54    JL=JL+1
      QUAN(1,K)=SRG
      QUAN(2,K)=GMASS
      IK=K
      NSRV=NSRV+1
      GO TO 56
55    NSRV=NSRV+1
      QUAN(1,NSRV)=SRG
      QUAN(2,NSRV)=GMASS
      IK=NSRV
56    IF (NSRV-99) 65,57,57
C     REDUCE S/R TABLE BY 20
57    DO 64 K=1,20
      NSRVM=NSRV-1
      SMALL=1.0
      AX=QUAN(2,2)-QUAN(2,1)
      AY=QUAN(1,2)-QUAN(1,1)
      AY=QUAN(1,2)-QUAN(1,1)
      A=SQRT(AX**2+AY**2)
      AX=AX/A
      AY=AY/A
      DO 62 KK=2,NSRVM
      BX=QUAN(2,KK+1)-QUAN(2,KK)
      BY=QUAN(1,KK+1)-QUAN(1,KK)
      B=SQRT(BX**2+BY**2)
      BX=BX/B
      BY=BY/B
      IF (BX) 58,61,58
```

```
58    IF (AX) 59,61,59
59    TMALL=ABS(AX*BY-AY*BX)
      IF (SMALL-TMALL) 61,61,60
60    SMALL=TMALL
      KKK=KK
61    AX=BX
62    AY=BY
      NSRV=NSRV-1
      DO 63 KK=KKK,NSRVM
      QUAN(1,KK)=QUAN(1,KK+1)
63    QUAN(2,KK)=QUAN(2,KK+1)
64    CONTINUE
65    IF (INCASE) 66,72,72
66    CALL PAGE (6+(NSRV+9)/10*3)
      WRITE (6,80) IK
67    IF (INCASE.GT.0) GO TO 72
      WRITE (6,82)
      DO 71 K=1,NSRV,10
      IF (NSRV-K-10) 68,69,69
68    L2=NSRV
      GO TO 70
69    L2=K+9
70    WRITE (6,83) (QUAN(2,II),II=K,L2)
      WRITE (6,84) (QUAN(1,II),II=K,L2)
71    CONTINUE
72    CONTINUE
      VAR1=SRG
      VAR2=GMASS
      GO TO 77
73    NSRVS=NSRV
C     STORE ALL S/R VALUES. CHARACT. LINE ACCEPTED
      DO 74 K=1,NSRVS
      SGMASS(K)=QUAN(2,K)
74    SSRG(K)=QUAN(1,K)
      GO TO 72
C     RESTORE ALL S/R VALUES
75    NSRV=NSRVS
      DO 76 L=1,NSRV
      QUAN(1,L)=SSRG(L)
76    QUAN(2,L)=SGMASS(L)
```

```
      GO TO 72
77    RETURN
C
C
78    FORMAT (38H AFTER LAST ENTRY IN S/R TABLE. GMASS=E13.6,22H,SR SET
     1TO LAST ENTRY,E13.6,22H,W/CORRESPONDING MASS E13.6)
79    FORMAT (40H BEFORE FIRST ENTRY IN S/R TABLE. GMASS=E13.6,23H SR SE
     1T TO FIRST ENTRY,E13.6,22H,W/CORRESPONDING MASS E13.6)
80    FORMAT (1H044X9HENTRY NO.I4,6H ADDED)
81    FORMAT (1H052XI3,20H ENTROPY VALUES READ)
82    FORMAT (7X1H112X1H212X1H312X1H412X1H512X1H612X1H712X1H812X1H911X2H
     110/5X4HMASS9(9X4HMASS)/6X3HS/R9(10X3HS/R)/)
83    FORMAT (1H0E12.6,9(E13.6))
84    FORMAT (10E13.6)
85    FORMAT (40H BEFORE FIRST ENTRY IN S/R TABLE.   S/R=E13.6,26H GMASS
     1 SET TO FIRST ENTRY,E13.6,15H,W/CORRES. S/R=E13.6)
86    FORMAT (39H BEYOND LAST ENTRY IN S/R TABLE.   S/R=E13.6,25H.GMASS
     1SET TO LAST ENTRY,E13.6,15H,W/CORRES. S/R=E13.6)
      END
```

```
      FUNCTION TSON (VM,G2,G3,G1)
      TSON=(G3*(VM**2)-(3.0-G1)+SQRT(G3*((G3*(VM**2)-2.0*(3.-G1))*(VM**2
     1)+G1+9.0)))/(4.0*G1*(VM**2))
      TSON=ATAN(SQRT(TSON)/SQRT(1.0-TSON))
      RETURN
      END
```

```
      FUNCTION DIF (XX,YY,M,N,L)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HCL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CO,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      MML=M-L
      TX=X(M-1,N)-X(M,N)
      TY=Y(M-1,N)-Y(M,N)
      IF (ABS(TX)+ABS(TY)) 1,1,2
    1 DIF=10.0
      GO TO 3
    2 DIF=SQRT(((XX-X(MML,N))**2+(YY-Y(MML,N))**2)/(TX**2+TY**2))
    3 RETURN
      END
```

```
      SUBROUTINE CHANGE (LA,LB)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      DIMENSION XH(100),YH(100),DFH(100),VMH(100),PH(100),SRH(100),IDH(
     1100)
C     LA=1,HOLD ALL ELEMENTS FROM FIRST DIM. BEGINING W/ELEMENT LB
C     LA=2,RETURN ALL STORED ELEMENTS TO FIRST DIM. BEG. W/ELEMENT LB
C     LA=3,SET THIRD DIM. TO SECOND DIM. ELEMENTS BEG. W/ELEMENT LB
C     LA=4,SET SECOND DIM. TO THIRD DIM. ELEMENTS BEG. W/ELEMENT LB
C     LA=5,RESET ALL SECOND DIM. ELEMENTS TO ZERO BEGINING W/ELEMENT LB
C     LA=6,STEP DOWN ALL FIRST DIM. ELEMENTS BY ONE BEG. W/ELEMENT LB
C     LA=7,SET ALL FIRST DIM. TO SECOND DIM. ELEMENTS BEG. W/ELEMENT LB
      GO TO (1,3,5,6,9,11,13), LA
1     DO 2 I=LB,NOW
      XH(I)=X(I,1)
      YH(I)=Y(I,1)
      DFH(I)=DF(I,1)
      VMH(I)=VM(I,1)
      PH(I)=P(I,1)
      SRH(I)=SR(I,1)
2     IDH(I)=ID(I,1)
      GO TO 14
3     DO 4 I=LB,NOW
      X(I,1)=XH(I)
      Y(I,1)=YH(I)
      DF(I,1)=DFH(I)
      VM(I,1)=VMH(I)
      P(I,1)=PH(I)
      SR(I,1)=SRH(I)
4     ID(I,1)=IDH(I)
      GO TO 14
5     J=3
      K=2
      GO TO 7
```

```
 6    J=2
      K=3
 7    DO 8 I=LB,NOW
      X(I,J)=X(I,K)
      Y(I,J)=Y(I,K)
      DF(I,J)=DF(I,K)
      VM(I,J)=VM(I,K)
      P(I,J)=P(I,K)
      SR(I,J)=SR(I,K)
 8    ID(I,J)=ID(I,K)
      GO TO 14
 9    DO 10 I=LB,NOW
      X(I,2)=0.0
      Y(I,2)=0.0
      DF(I,2)=0.0
      VM(I,2)=0.0
      P(I,2)=0.0
      SR(I,2)=0.0
10    ID(I,2)=0
      GO TO 14
11    NOWM=NOW-1
      DO 12 I=LB,NOWM
      X(I,1)=X(I+1,1)
      Y(I,1)=Y(I+1,1)
      DF(I,1)=DF(I+1,1)
      VM(I,1)=VM(I+1,1)
      P(I,1)=P(I+1,1)
      SR(I,1)=SR(I+1,1)
12    ID(I,1)=ID(I+1,1)
      GO TO 14
13    J=1
      K=2
      GO TO 7
14    RETURN
      END
```

```
      SUBROUTINE INPUT (LA,LB,LC)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      GO TO (1,2), LA
    1 N=LB
      GO TO 3
    2 N=NOW
    3 DO 6 I=LB,N
      READ (3) X(I,LC),Y(I,LC),DF(I,LC),VM(I,LC),SR(I,LC)
      P(I,LC)=((1.0+GF(4)*VM(I,LC)**2)**(-GF(10)))*SR(I,LC)
      IF (DET) 4,5,4
    4 X(I,LC)=(X(I,LC)-DET)*R/RFIT-HOL(41)
      Y(I,LC)=Y(I,LC)*R/RFIT
      IF( (MID.EQ.1) .OR. (HOL(37).EQ.0.0 ) ) GO TO 5
      XP=X(I,LC)-R+HOL(41)
      YP=Y(I,LC)
      XPP=XP*HOL(32)-YP*HOL(33)
      YPP=XP*HOL(33)+YP*HOL(32)
      X(I,LC)=XPP+R-HOL(41)
      Y(I,LC)=YPP
    5 DF(I,LC)=DF(I,LC)/RAD
      DF(I,LC)=DF(I,LC)+HOL(37)
    6 CONTINUE
      RETURN
C
      END
```

```
      SUBROUTINE SHAPE
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      COMMON /BETA/ D(102)
      DIMENSION NUMBER(3)
      EM=TAN(HOL(23))
      EM2=EM*EM
      BL=D(5)
      R=1.0
      XB(1)=(1.0-EM/SQRT(D(5)+EM2))/D(5)
      YB(1)=SQRT((2.0-D(5)*XB(1))*XB(1))
      SLOPE2(1)=ATAN(EM)*RAD
      J1(1)=0
      J1(2)=1
      HOL(4)=HOL(3)
      HOL(5)=HOL(3)
      NSEG=2
      NSEGM=NSEG
      NSEG=NSEG+1
      XF(1)=0.0
      XB(2)=2.0
      SLOPE1(2)=EM
      SLOPE1(3)=EM
      DO 5 I=2,NSEG
      IF (I-NSEG) 2,1,1
    1 J1(I)=1
      XB(I)=1.1*XB(I-1)
      GO TO 3
    2 J=I+2
    3 XF(I)=XB(I-1)
      YF(I)=YB(I-1)
      YB(I)=YF(I)+SLOPE1(I)*(XB(I)-XF(I))
      SLOPE2(I)=SLOPE1(I)
      AB(I)=0.0
```

```
      AC(I)=0.0
      AD(I)=SLOPE1(I)
      AE(I)=YF(I)
      IF (I-NSEGM) 5,4,5
4     SLOPE1(NSEG)=SLOPE2(NSEGM)
5     CONTINUE
      DO 7 I=3,J
      DX=XB(I-2)-XF(I-2)
      NUMBER(I-2)=DX/(HOL(39)*HOL(I))
      FIGURE=NUMBER(I-2)
      IF (FIGURE) 7,7,6
6     HOL(I)=DX/(HOL(39)*FIGURE)
7     CONTINUE
      HOL(J+1)=HOL(J)
      WRITE (6,18)
      DO 11 I=1,NSEGM
      K=I+2
      IF (I-1) 9,9,8
8     SLOPE1(I)=ATAN(SLOPE1(I))*RAD
      SLOPE2(I)=SLOPE1(I)
      GO TO 10
9     CALL PAGE (2)
      WRITE (6,12)
      WRITE (6,14) I,XB(I),YB(I),HOL(K)
      WRITE (6,15) R,D(5)
      GO TO 11
10    CALL PAGE (3)
      WRITE (6,13)
      WRITE (6,14) I,XB(I),YB(I),HOL(K)
      WRITE (6,16) SLOPE1(I),SLOPE2(I)
      WRITE (6,17) AB(I),AC(I),AD(I),AE(I)
11    CONTINUE
      SLOPE1(NSEG)=ATAN(SLOPE1(NSEG))*RAD
      SLOPE2(1)=0.0
      RETURN
C
C
12    FORMAT (15H  BLUNT SEGMENT)
13    FORMAT (15H  CUBIC SEGMENT)
14    FORMAT (1H+,14XI3,2X3HX2=E13.6,3X3HY2=E13.6,3X3HX*=E13.6)
```

```
15    FORMAT (1H 29X2HR=E13.6,3X2HB=E13.6)
16    FORMAT (1H 29X8HSLOPE 1=F10.5,4H DEG,3X8HSLOPE 2=F10.5,4H DEG)
17    FORMAT (1H 29X2HA=E13.6,5X2HB=E13.6,5X2HC=E13.6,5X2HD=E13.6)
18    FORMAT (1H )
      END
```

```
      SUBROUTINE DROP (I,J,K,L)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      IF (TIN) 1,1,2
1     IF (I-1) 3,3,2
2     IF (L-I) 18,18,11
3     L=I
4     IZ=I
      JZ=J
      MZ=L
      LZ=J-1
      CMINUS=DF(IZ,JZ)-ANGLE(VM(IZ,JZ))-0.02
      CPLUS=DF(MZ,LZ)+ANGLE(VM(MZ,LZ))+0.02
      DELX=X(IZ,JZ)-X(MZ,LZ)
      DELY=Y(IZ,JZ)-Y(MZ,LZ)
      IF (DELX) 6,5,6
5     GEO=-HOL(63)
      GO TO 8
6     GEO=ATAN(DELY/DELX)
      IF (DELX) 8,8,7
7     GEO=GEO-HOL(62)
8     IF ((CMINUS-GEO)) 11,11,9
9     GEP=GEO+HOL(62)
      IF (GEP-CPLUS) 11,11,10
10    K=1
      GO TO 18
11    NSKIP=NSKIP+1
      K=2
      IF (TIN-1.0) 12,12,13
C     HOLD ALL ELEMENTS FROM FIRST DIM. BEGINING W/ELEMENT TWO
12    CALL CHANGE (1,1)
C     STEP DOWN ALL ELEMENTS FROM DIM. ONE BEGINING W/ELEMENT TWO
13    CALL CHANGE (6,2)
      IF (TIN) 15,15,14
```

```
14     IHOP=HOL(45)
       TIN=1+(1+IHOP)/2-HOL(46)
15     NOW=NOW-1
       IF (NOW-1) 17,17,16
16     IF (TIN.LE.0.) GO TO 4
       L=L-1
       GO TO 2
17     KICK=1
18     IF (INCASE-4) 20,19,19
19     WRITE (6,21) I,J,K,TIN,NOW,NSKIP,GEO,CMINUS,CPLUS
20     RETURN
C
C
21     FORMAT (14H DROP PROG. I=I3,3H J=I3,3H K=I3,5H TIN=E13.6,5H NOW=I3
      1,7H NSKIP=I3/5H GEO=E13.6,5H CMI=E13.6,5H CPL=E13.6)
       END
```

```
      SUBROUTINE OUTPUT (LA,LB,LC)
C     IDEAL GAS OUTPUT SUBROUTINE
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      DIMENSION DF3(100)
      DATA Q001HL/6H P/PTI/
      CALL SLITET (1,K000FX)
      GO TO (6,1), K000FX
1     GO TO (2,4,6), LGO
2     IF (NSUM-LONE) 6,3,3
3     LSAVE=INCASE
      INCASE=4
      LGO=2
      GO TO 6
4     IF (NSUM-LTWO) 6,6,5
5     INCASE=LSAVE
      LGO=3
6     GO TO (7,11,12,13,21,14,14,22), LA
7     REWIND 4
      CALL PAGE (4)
      LONE=HOL(50)
      LTWO=HOL(49)
      IF (HOL(50)) 8,8,9
8     LGO=3
      GO TO 10
9     LGO=1
10    TYPE=(+Q001HL)
      GO TO 25
11    CALL PAGE (6)
      WRITE (6,28) PFS,VFS,GF(1),REF
      GO TO 25
12    CALL PAGE (6)
      WRITE (6,31) THP,TYPE
      GO TO 14
```

```
13    CALL PAGE (6)
      WRITE (6,35) NSUM,THP,TYPE
14    DO 17 I=LB,NOW
      DFP=DF(I,LC)*RAD
      PR=(P(I,LC)-PFS)/RV2
      IF (LA-6) 15,18,19
15    CALL PAGE (1)
      WRITE (6,29) I,X(I,LC),Y(I,LC),DFP,VM(I,LC),PR,SR(I,LC)
      IF (ID(I,LC)) 16,17,16
16    WRITE (6,36) ID(I,LC)
17    CONTINUE
      CALL PAGE (3)
      WRITE (6,32) CD
      WRITE (6,26) HOL(57)
      NSUM=NSUM+1
      GO TO 25
18    IF (INCASE.LT.4) GO TO 25
      CALL PAGE (4)
      WRITE (6,33) X(I,LC),Y(I,LC),DFP,VM(I,LC),TYPE,PR,SR(I,LC)
      GO TO 25
19    CONTINUE
      CALL PAGE (1)
      WRITE (6,34) X(I,LC),Y(I,LC),DFP,VM(I,LC),TYPE,PR,SR(I,LC)
      IF (LC-1) 25,25,20
20    CALL PAGE (1)
      WRITE (6,27) HOL(57)
      GO TO 25
21    CALL PAGE (4)
      WRITE (6,30) NSUM,TYPE
      GO TO 14
22    CONTINUE
      DO 23 I=1,NOW
      DF3(I)=DF(I,LC)*RAD
23    CONTINUE
      WRITE (4) NSUM,NOW
      DO 24 J=1,NOW
24    WRITE (4) X(J,LC),Y(J,LC),DF3(J),VM(J,LC),SR(J,LC)
25    IF (INCASE.LT.2.AND.(LB.EQ.NOW.AND.LA.EQ.7)) NSUM=NSUM+1
      RETURN
C
```

```
26    FORMAT (1H+,60X20H(DELTA M)/(M TOTAL)=E15.6/1H 119(1H-))
27    FORMAT (5X,20H(DELTA M)/(M TOTAL)=E15.6)
28    FORMAT (36H0----FREE STREAM AND CONSTANT VALUES84(1H-)/12H0 (P/PT)
     1INF=E15.6/12H    MACH NO =0PF9.5/5X7HGAMMA =F9.5/5X7HR-REF =E15.6/
     21X,119(1H-))
29    FORMAT (2X,I3,2G15.6,2G12.5,3G14.6)
30    FORMAT (18H0----C-SUMMARY NO.I3,99(1H-)/6H0POINT7X1HX14X1HY11X5HDE
     1LTA,7X,4HMACH,7X,A6,9X,6HPT/PT0,/)
31    FORMAT (28H0----INITIAL VALUE LINE DATA92(1H-)/14X12HTHETA SHOCK=
     1E13.6/6H0POINT7X1HX14X1HY11X5HDELTA7X4HMACH7XA6,9X6HPT/PT0/)
32    FORMAT (26H0                DRAG COEF.=E13.6)
33    FORMAT (28H0----INITIAL VALUE LINE DATA92(1H-)/5X2HX=E13.6,4H  Y=
     1E13.6,8H  DELTA=0PF10.5,7H  MACH=F10.5,2X,A6,1H=E13.6,9H  PT/PT0=
     2E13.6/1X,119(1H-))
34    FORMAT (5X2HX=E13.6,2X2HY=E13.6,2X6HDELTA=0PF10.5,2X5HMACH=0PF10.5
     1,2XA6,1H=E13.6,9H  PT/PT0=E13.6)
35    FORMAT (18H0----C-SUMMARY NO.I3,99(1H-)/14X12HTHETA SHOCK=E13.6/6H
     10POINT,7X,1HX,14X,1HY,11X,5HDELTA,7X,4HMACH,7X,A6,9X,6HPT/PT0,/)
36    FORMAT (1H+101X12)
      END
```

```
      SUBROUTINE WAVE (DELTA,DELTO,QM,P1,S1,LIM)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      QM2=QM**2
      IF (INCASE-4) 2,1,1
    1 WRITE (6,15) DELTA,DELTO,LIM
    2 GO TO (9,3), LIM
    3 TH1=DELTO+ANGLE(QM)
      D1=0.0
      IW=0
      TH2=TSON(QM,GF(2),GF(3),GF(1))+DELTO
    4 SITH=SIN(TH2-DELTO)
      VMN=QM*SITH
      VMN2=VMN**2
      DW=ATAN((2.0*(VMN2-1.0)/TAN(TH2-DELTO))/(2.0+GF(3)*(QM2)-2.0*VMN2)
     1)+DELTO
      IF (ABS(DELTA/DW-1.0)-.000001) 5,5,6
    5 THOTA=TH2-DELTO
      HOL(68)=TH2
      GO TO 10
    6 IW=IW+1
      IF (IW-50) 8,8,7
    7 WRITE (6,13)
      KICK=1.0
      GO TO 12
    8 TH3=TH2
      TH2=((TH2-TH1)*(DELTA-D1)/(DW-D1))+TH1
      D1=DW
      TH1=TH3
      GO TO 4
    9 THETA=DELTA
      THOTA=THETA-DELTO
      SITH=SIN(THOTA)
      VMN=QM*SITH
```

```
      VMN2=VMN**2
10    PR=((GF(9)*VMN2-GF(2))/GF(3))
      PW=P1*PR
      DW=ATAN((2.0*(VMN2-1.0)/TAN(THDTA))/(2.0+GF(3)*QM2-2.0*VMN2))+
     1DELTO
      SW=S1*((GF(3)*VMN2/(GF(2)*VMN2+2.0))**GF(10))*((PR)**(-GF(11)))
      VMW=(GF(8)*VMN2*(QM2)-4.0*(VMN2-1.0)*(GF(1)*VMN2+1.0))/((2.0*GF(1)
     1*VMN2-GF(2))*(GF(2)*VMN2+2.0))
      VMW=SQRT(VMW)
      IF (INCASE-4) 12,11,11
11    WRITE (6,14) PW,DW,SW,VMW
12    RETURN
C
C
13    FORMAT (5X42HFAILED TO CONVERGE SHOCK ANGLE IN 50 TRIES)
14    FORMAT (10X3HPW=E15.6,5X3HDW=E15.6,5X3HSW=E15.6,5X4HVMW=E15.6)
15    FORMAT (9H IN WAVE/5X6HDELTA=E15.6,5X6HDELTO=E15.6,5X4HLIM=I5)
      END
```

```
      SUBROUTINE SHOCK (IRS,JRS)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      DIMENSION RESID(2),TH(2),TAN2S(99)
      IDROP=0
      G1=GF(1)
      I=IRS
      J=JRS
      JP1=J+1
      NS=ID(I,J)
      TWO=2.0
      NHOL=-1
      IT=NOW-1
      DO 1 K=1,IT
      TAN2S(K)=0.0
1     IF (ABS(Y(K,J)-Y(K+1,J))+ABS(X(K,J)-X(K+1,J)).NE.0.0) TAN2S(K)=(Y(
     1K,J)-Y(K+1,J))/(X(K,J)-X(K+1,J))
      IF (INCASE-4) 3,2,2
2     WRITE (6,84) ((K,TAN2S(K)),K=1,NOW)
3     IT=0
      IT1=0
      IWAY=1
      ISS=0
      D2=DF(I,J)
      AMU2=ANGLE(VM(I,J))
      THO=THS(1,1)
      IF (INCASE-4) 4,5,5
4     IP=1
      GO TO 6
5     IP=2
6     N=1
      IHOW=HOL(46)
      NHOL=I*IHOW-1
      DO=HOL(37)
```

```
      SO=1.0
      IF (HOL(46)) 8,8,7
7     XU=X(I,J)
      YU=Y(I,J)
      X4=X(I+1,J)
      Y4=Y(I+1,J)
      D4=DF(I+1,J)
      P4=P(I+1,J)
      VM4=VM(I+1,J)
      AMU4=ANGLE(VM4)
      TAN3=TAN(D4+HOL(45)*AMU4)
      IDROP=I+1
      GO TO 19
8     IF (INCASE-4) 10,9,9
9     WRITE (6,73) HOL(53),HOL(54),HOL(69),XSTAR,XSTARB
10    IF (HOL(54)-1.0) 13,11,13
11    XSTAR=HOL(53)*0.5
      IF (INCASE-4) 13,12,12
12    WRITE (6,86) XSTAR
13    SM=XSTAR
      IF (INCASE-4) 15,14,14
14    WRITE (6,87) XSTAR
15    IF (SM) 16,16,18
16    WRITE (6,72) SM,XSTAR,AMU2,D2,THS(NS,J),NS,J
17    KICK=KICK+1
      GO TO 69
18    SAVS=SIN(THS(NS,J))
      SAVC=COS(THS(NS,J))
      XU=SM*(SAVC-SAVS)+X(I,J)
      YU=SM*(SAVC+SAVS)+Y(I,J)
      TANC=-SAVC/SAVS
19    THS3=TSONIC
      D3=DSONIC
      P3=PSONIC
      VM3=VMSON
      AMU3=BSONIC
      GO TO (21,20), IP
20    WRITE (6,78) NS,J,THS(NS,J),XU,YU,D3,P3,THS3,AMU3
21    TANAV=TAN((THO+THS3)*0.5)
      IF (NHOL) 23,22,23
```

```
22    X3=(YU-Y4-XU*TANAV+X4*TAN3)/(TAN3-TANAV)
      Y3=((X4-XU)*(TAN3*TANAV)-Y4*TANAV+YU*TAN3)/(TAN3-TANAV)
      GO TO 24
23    X3=(YU-Y(I,J)-XU*TANC+X(I,J)*TANAV)/(TANAV-TANC)
      Y3=YU+TANC*(X3-XU)
24    FMO=VFS
      PO=(1.0+GF(4)*(FMO**2))**(-GF(10))
      THSAVE=ANGLE(FMO)
      GO TO (26,25), IP
25    WRITE (6,79) X3,Y3,TANAV
26    TAN2=TAN2S(I)
      IF (NHOL) 27,28,27
27    TAN3=TAN(.5*(D3+D2+(AMU3+AMU2)*HOL(45)))
28    IF (NHOL) 29,54,29
29    L=I
      TAN2=TAN2S(L)
30    X4=(Y(L,J)-Y3+X3*TAN3-X(L,J)*TAN2)/(TAN3-TAN2)
      Y4=Y(L,J)+TAN2*(X4-X(L,J))
      IDROP=L
      GO TO (32,31), IP
31    WRITE (6,80) X4,Y4,IT1,X(L,J),Y(L,J),TAN2,TAN3,L,J
32    IGATE=0
      NGATE=0
      DEL3=DIF(X4,Y4,L+1,J,1)
      IF (DEL3-1.0) 38,38,33
33    L=L+1
      IF (L-NOW) 35,34,34
34    XSTAR=XSTAR*0.5
      ISS=ISS+1
      IF (ISS-3) 18,18,17
35    IGATE=1
      IF (NGATE) 37,37,36
36    WRITE (6,74)
      GO TO 17
37    TAN2=TAN2S(L)
      GO TO 30
38    DEL1=DIF(X4,Y4,L+1,J,0)
      IF (DEL1-1.0) 40,40,39
39    WRITE (6,75)
      GO TO 17
```

```
40    K=J
      LL=L+1
      IF (DEL3-1.0) 42,42,41
41    L=L+1
      TAN2=TAN2S(L-1)
      WRITE (6,85) DEL3
      GO TO 17
42    D4=DF(L,J)+DEL3*(DF(LL,K)-DF(L,J))
      P4=P(L,J)+DEL3*(P(LL,K)-P(L,J))
      VM4=VM(L,J)+DEL3*(VM(LL,K)-VM(L,J))
      AMU4=ANGLE(VM4)
      TAN3=TAN((D4+D3+(AMU4+AMU3)*HOL(45))*.5)
      GO TO (44,43), IP
43    WRITE (6,81) D4,P4,VM4,AMU4,TAN3,IT
44    IF (IT1) 45,45,47
45    IT1=1
46    X4A=X4
      Y4A=Y4
      D4A=D4
      P4A=P4
      VM4A=VM4
      GO TO 28
47    IT1=IT1+1
      IF (IT1-50) 49,49,48
48    WRITE (6,76) X3,Y3,NSUM
      X4=0.5*(X4+X4A)
      Y4=0.5*(Y4+Y4A)
      D4=0.5*(D4+D4A)
      P4=0.5*(P4+P4A)
      VM4=0.5*(VM4+VM4A)
      GO TO 54
49    IF (ABS(X4A/X4-1.0)-TEST) 50,50,46
50    IF (ABS(Y4A/Y4-1.0)-TEST) 51,51,46
51    IF (ABS(D4A/D4-1.0)-TEST) 52,52,46
52    IF (ABS(P4A/P4-1.0)-TEST) 53,53,46
53    IF (ABS(VM4A/VM4-1.0)-TEST) 54,54,46
54    AMAV=(AMU3+AMU4)*0.5
      DAV=(D3+D4)*0.5
      C1=HOL(45)*(D3-D4)+DEM*(SIN(DAV)*SIN(AMAV)*SQRT((X3-X4)**2+(Y3-Y4)
     1**2)/(.5*(Y3+Y4)))
```

```
      C2=G1/SIN(2.0*AMAV)
      RT=2.0*C1*C2
      RESID(IWAY)=ALOG(P3/P4)+RT
      TH(IWAY)=THS3
      IT1=0
      GO TO (56,55), IP
   55 WRITE (6,82) AMAV,DAV,C1,C2,RESID(IWAY),TH(IWAY),IWAY
   56 GO TO (57,58), IWAY
   57 IT=1
      IWAY=2
      THS3=THS(NS,J)
      D3=DF(I,J)
      P3=P(I,J)
      AMU3=AMU2
      VM3=VM(I,J)
      S3=SR(I,J)
      GO TO 21
   58 SHINC=TH(1)-TH(2)
      CORR=RESID(1)*(SHINC/(RESID(2)-RESID(1)))
      RATIO=CORR/SHINC
      IF (ABS(RATIO)-2.0) 60,60,59
   59 CORR=SHINC*SIGN(TWO,RATIO)
   60 THS3=TH(1)+CORR
      IF ((THS3-THSAVE)*HOL(45)) 61,61,62
   61 THS3=THSAVE
   62 GO TO (64,63), IP
   63 WRITE (6,83) TH(1),RESID(1),TH(2),RESID(2),THS3,IT
   64 TH(1)=TH(2)
      RESID(1)=RESID(2)
      IT=IT+1
      IF (IT-75) 67,67,65
   65 WRITE (6,77)
      GO TO 17
   66 CALL WAVE (THS3,DO,FMO,PO,SO,1)
      D3=DW
      P3=PW
      VM3=VMW
      AMU3=ANGLE(VM3)
      IF (HOL(46).GT.0.0) TAN3=TAN(0.5*(D3+D4+(AMU3+AMU4)*HOL(45)))
      S3=SW
```

```
      GO TO 21
67    IF (ABS(RESID(2))-TESU) 68,66,66
68    HOL(60)=N
      K=N
      X(K,JP1)=X3
      Y(K,JP1)=Y3
      IHOG=HOL(45)
      ID(K,JP1)=NS2+(1+IHOG)/2
      DF(K,JP1)=D3
      P(K,JP1)=P3
      VM(K,JP1)=VM3
      SR(K,JP1)=S3
      THS(NS2+1,JP1)=THS3
      DELY=Y3*COS(HOL(37))-(X3-R)*SIN(HOL(37))
      VMASS(1)=(DELY**(NID+1))/HOL(58)
C     INSERT TEMPORARY SR VALUE INTO TABLE FROM BOW WAVE
      CALL ENTROP (SR(1,2),VMASS(1),4,0)
69    ICALL=1
      IF (HOL(61)) 70,70,71
70    CALL DROP (ICALL,J,IDMY,IDROP)
71    RETURN
C
C
C
72    FORMAT (5X32HSM IS ZERO OR NEGATIVE IN RSHOCK/10X3HSW=E15.6/15X6HX
     1STAR=E15.6/20X5HAMU2=E15.6/25X3HD2=E15.6/30X9HTH(NS,J)=E15.6/35X3H
     2NS=I5/40X2HJ=I5)
73    FORMAT (5X8HHOL(53)=E15.6/5X8HHOL(54)=E15.6/5X8HHOL(69)=E15.6/7X6H
     1XSTAR=E15.6/6X7HXSTARB=E15.6//)
74    FORMAT (5X35HOSCILLATED THROUGH NGATE THEN IGATE)
75    FORMAT (5X35HOSCILLATED THROUGH IGATE THEN NGATE)
76    FORMAT (5X34HEXCEEDED 50 ITERATIONS FOR POINT 4,5X3HX3=E15.7,5X3HY
     13=E15.7,5X3HC-=I4)
77    FORMAT (5X32HEXCEEDED 75 ITERATIONS FOR SHOCK)
78    FORMAT (10X7HNO. 202/11X3HNS=I5,5X2HJ=I5/15X10HTHS(NS,J)=E15.6/
     120X3HXU=E15.6/25X3HYU=E15.6/30X3HD3=E15.6/30X3HP3=E15.6/40X5HTHS3=
     2E15.6/45X5HAMU3=E15.6)
79    FORMAT (20X3H206/15X3HX3=E15.6/20X3HY3=E15.6/25X6HTANAV=E15.6)
80    FORMAT (5X3HX4=E15.8,5X3HY4=E15.8,5X4HIT1=I5/10X7HX(L,J)=E15.8,5X7
     1HY(L,J)=E15.8,5X5HTAN2=E15.8,5X5HTAN3=E15.8/10X,2HL=I5,5X2HJ=I5)
```

```
81     FORMAT (10X3H213/15X3HD4=E15.6/20X3HP4=E15.6/25X4HVM4=E15.6/35X5HA
      1MU4=E15.6/40X5HTAN3=E15.6/45X3HIT=I5)
82     FORMAT (10X3H215/15X5HAMAV=E15.6/20X4HDAV=E15.6/25X3HC1=E15.6/30X3
      1HC2=E15.6/35X12HRESID(IWAY)=E15.6/40X9HTH(IWAY)=E15.6/45X5HIWAY=I5
      2)
83     FORMAT (10X3H217/10X6HTH(1)=E15.6,5X9HRESID(1)=E15.6/15X6HTH(2)=
      1E15.6,5X9HRESID(2)=E15.6/20X5HTHS3=E15.6,10X3HIT=I5)
84     FORMAT (10X1HK,7X8HTAN2S(K)/(I11,E15.6))
85     FORMAT (5X18HFAILED AT 68,DEL3=E15.6)
86     FORMAT (5X16HAFTER 244,XSTAR=E15.6)
87     FORMAT (5X16HAFTER 243,XSTAR=E15.6)
       END
```

```
      SUBROUTINE MESH (IGO)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUD,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      IT=0
      IGOGO=0
      IF (INCASE-4) 2,1,1
    1 WRITE (6,43)
      WRITE (6,44) XSTAR,XSTARB,XLAST,XSPEC(ILK),XSAVE,DX1,XINTP,DXSTAR,
     1HOL(48),HOL(54),HOL(69),XS2,DXRATO,IXS,IGO,LOOP,ISPEC,ILK
    2 DXSTAR=X(NOW,2)-XLAST
      IF (HOL(69)) 3,3,24
    3 IF (LOOP) 4,4,5
    4 IXS=0
      XS2=XSTAR
      DX2=X(NOW,2)
    5 IF (DXSTAR) 6,6,8
    6 WRITE (6,40) DXSTAR
    7 IGO=2
      GO TO 30
    8 LOOP=LOOP+1
      IF (LOOP-25) 10,10,9
    9 WRITE (6,41) XSTAR,XSAVE,DXRATO,DX1,XSTARB
      GO TO 7
   10 IF (ISPEC) 15,15,11
   11 DXRATO=DXSTAR/DX1
      IF (ABS(DXRATO-1.)-.0005) 23,12,12
   12 IF (IXS-2) 13,20,20
   13 IXS=IXS+1
      XS2=XSTAR
      DX2=X(NOW,2)
      XSTAR=XSTAR/DXRATO
   14 NOW=NOW-1
      IGO=1
      GO TO 30
```

```
15    IF (DXSTAR-XSTARB) 16,16,17
16    DXRATO=DXSTAR/XSTARB
      IF (DXRATO-HOL(38)) 17,24,24
17    IF (IXS) 18,18,19
18    IXS=1
      XSTAR=0.9*XSTAR*XSTARB/DXSTAR
      GO TO 14
19    XINTP=XLAST+HOL(39)*XSTARB
20    DX3=X(NOW,2)
      XST=XS2+(XSTAR-XS2)*(XINTP-DX2)/(X(NOW,2)-DX2)
      IF (XST) 22,22,21
21    XSTAR=XST
      GO TO 14
22    XSTAR=0.5*XS2
      WRITE (6,42) XSTAR,XS2
      DX2=DX3
      GO TO 14
23    ISPEC=0
      XSTARB=HOL(ILK+3)
      IGOGO=1
      XSPEC(ILK)=0.0
      HOL(40)=ILK
      XSTAR=XSAVE
      IT=1
24    DO 25 ILK=1,NSEG
      IF (XSPEC(ILK)) 25,25,33
25    CONTINUE
26    IF (HOL(54)) 27,27,28
27    XSTAR=HOL(39)*XSTAR/DXRATO
28    ILK=HOL(40)
      HOL(40)=0.0
      IGO=3+IGOGO
29    XLAST=X(NOW,2)
30    IF (INCASE-4) 32,31,31
31    WRITE (6,44) XSTAR,XSTARB,XLAST,XSPEC(ILK),XSAVE,DX1,XINTP,DXSTAR,
     1HOL(48),HOL(54),HOL(69),XS2,DXRATO,IXS,IGO,LOOP,ISPEC,ILK
32    RETURN
33    XMX=XSPEC(ILK)-X(NOW,2)
      IF (XMX) 34,34,35
34    LGO=1
```

```
      XPAST=XLAST
      GO TO 37
35    CONTINUE
      IF (XMX-XSTARB*1.2) 36,36,26
36    LGO=2
      XPAST=X(NOW,2)
37    ISPEC=1
      HOL(69)=0.0
      ILK=ILK
      XSAVE=XSTAR
      DX1=XSPEC(ILK)-XPAST
      XSTAR=DX1*XSTAR/DXSTAR
      XINTP=XSPEC(ILK)
      GO TO (14,38), LGO
38    IGO=3+IGOGO
      IF (HOL(40)) 29,29,39
39    ISV=ILK
      ILK=HOL(40)
      HOL(40)=ISV
      GO TO 29
C
C
C
40    FORMAT (44HONEG. BODY INCREMENT--PROGRAM STOPPED--INC.=E13.6)
41    FORMAT (40HOCANNOT CONVERGE TO SPECIAL BODY POINT  /5X6HXSTAR=
     1E13.6/10X6HXSAVE=E13.6/15X8HDXRATIO=E13.6/20X4HDX1=E13.6/25X7HXSTA
     2RB=E13.6)
42    FORMAT (1X6HXSTAR=E13.7,5X4HXS2=E13.7)
43    FORMAT (1X10(1H,),21HINITIAL VALUE IN MESH)
44    FORMAT (7E18.6/6E18.6/6I18)
      END
```

```
      SUBROUTINE PAGE (LINE)
      COMMON /FLOAT/ X(100,2),Y(100,2),P(100,2),DF(100,2),VM(100,2),SR(
     1100,2),GF(15),VMASS(100),SLOPE1(3),SLOPE2(3),XF(3),YF(3),XB(3),YB(
     23),AB(3),AC(3),AD(3),AE(3),THS(2,2),HOL(72),XSPEC(3),XSTAR,DEM,PFS
     3,VMSON,RAD,XLAST,AMUO,VMW,TSONIC,PO,TEST,R,DFP,BSONIC,PW,FMO,THP,
     4VFS,TYPE,TESU,BL,PSONIC,XSTARB,REF,RFIT,EL,TIN,DSONIC,RV2,CD,DET,
     5DW,SW
      COMMON /INTEG/ J1(3),NSEG,KPR,NSKIP,IPUNCH,NSUM,IT,ISEG,LOOP,MID,
     1KICK,NOW,ILK,NS2,ID(100,2),INCASE,ISPEC
      IF (LINE) 1,1,2
    1 IPAGE=0
      LPP=52
      NLINE=0
      GO TO 5
    2 IF (INCASE-2) 3,3,8
    3 IF (NLINE+LINE-LPP) 4,4,5
    4 NLINE=NLINE+LINE
      GO TO 8
C     RESTORE CARRIAGE AND PRINT PAGE HEADING
    5 IPAGE=IPAGE+1
      WRITE (6,9) IPAGE
      IF (LINE-LPP) 7,6,6
    6 NLINE=LINE-LPP+3
      GO TO 8
    7 NLINE=LINE+3
    8 RETURN
C
    9 FORMAT (1H1,111X,4HPAGE,I4)
      END
```

```
      FUNCTION ANGLE (X)
C
      ANGLE=ASIN(1.0/X)
      RETURN
      END
```

```
      FUNCTION VELOC (VM,G4)
      VELOC=G4*(VM**2)
      VELOC=SQRT(VELOC/(1.0+VELOC))
      RETURN
      END
```

```
      OVERLAY (BLUNT,4,0)
      PROGRAM IVSES
      COMMON /BETA/ D(7),XM,GAMA,ALFA,RAMDA,RINGS,E(89),MERID,BDRAG
      COMMON /TRANS/ LO,L1,LL,MS
      DIMENSION X(42,44),Y(42,44),DF(42,44),VM(42,44),SR(42,44),NO(40),
     1PSI(19),YH(40),DFH(40),VMH(40),SRH(40),RP(19,20),TH(19,20),PT(19,
     2 20),XH(19,20),VV(19,20)
      REWIND 4
      RAD=57.295779513
      NRING=RINGS
      SINA=SIN(ALFA)
      COSA=COS(ALFA)
      TANA=SINA/COSA
      ALDEG=ALFA*RAD
      RADEG=RAMDA*RAD
      X1=1.-SIN(RAMDA)
      WRITE (6,11) ALDEG,RADEG,X1
      N=(LL-LO)/40+1
      M=LL/40+1
      M2=(LL-LO+1)/M+2
      MX=0
      LX=0
      KX=1
      DO 4 J=1,LL
      K=J-LO+2
      JX=0
      READ (4) NSUM,NOW
      DO 3 I=1,NOW
      READ (4) A1,A2,A3,A4,A5
      L=J-MS
      IF (I.GT.1.OR.(L/M)*M.NE.L.OR.(L.LT.1)) GO TO 1
      MX=MX+1
      X(MX,M2)=A1
      Y(MX,M2)=A2
      DF(MX,M2)=A3
      VM(MX,M2)=A4
      SR(MX,M2)=A5
    1 IF (I.NE.NOW.OR.J.GT.L1) GO TO 2
      LX=LX+1
      X(LX,1)=A1
```

```
      Y(LX,1)=A2
      DF(LX,1)=A3
      VM(LX,1)=A4
      SR(LX,1)=A5
2     IF (K.LT.2.OR.(K/N)*N.NE.K) GC TO 3
      IF (JX.EQ.0) KX=KX+1
      IF (JX.EQ.0) JX=1
      X(I,KX)=A1
      Y(I,KX)=A2
      DF(I,KX)=A3
      VM(I,KX)=A4
      SR(I,KX)=A5
      NO(KX)=NOW
3     CONTINUE
4     CONTINUE
      NO(1)=LX
      NO(M2)=MX
      MERIDS=MERID*2-1
      FLOATM=MERIDS-1
      DA=180./FLOATM
      COSO=1.
      SINO=0.
      DO 9 I=1,MERIDS
      XI=I-1
      PSI(I)=DA*XI-90.
      COS1=1./SQRT(1.+(SIN(PSI(I)/RAD)*TANA)**2)
      SIN1=-SIN(PSI(I)/RAD)*TANA*COS1
      ANG=ASIN(SIN1)*RAD
      SIN2=SIN1*COSO-COS1*SINO
      COS2=COS1*COSO+SIN1*SINO
      M=0
      DO 7 J=1,M2
      NOW=NO(J)
      IF (NOW.LT.3.AND.(J.NE.1.OR.J.NE.M2)) GO TO 7
      IF (NOW.LT.2) GO TO 7
      DO 5 K=1,NOW
5     CALL ROTATE (X(K,J),Y(K,J),SIN2,COS2,1.0)
      IF (J.EQ.1.OR.J.EQ.M2) GO TO 6
      IF (X(1,J).GE.X1.OR.X(NOW,J).LE.X1) GO TO 7
6     M=M+1
```

```
      CALL SECOND (X(1,J),Y(1,J),NOW,X1,YH(M))
      IF (M.EQ.1) YH(1)=SQRT(2.0*X1-X1*X1)
      CALL SECOND (X(1,J),DF(1,J),NOW,X1,DFH(M))
      IF (M.EQ.1) DFH(1)=ATAN((1.0-X1)/YH(1))*RAD+ANG
      CALL SECOND (X(1,J),VM(1,J),NOW,X1,VMH(M))
      CALL SECOND (X(1,J),SR(1,J),NOW,X1,SRH(M))
    7 CONTINUE
      DR=(YH(M)-YH(1))/(RINGS-1.)
      DO 8 J=1,NRING
      XJ=J-1
      XH(I,J)=X1
      RP(I,J)=YH(1)+XJ*DR
      CALL SECOND (YM,DFH,M,RP(I,J),TH(I,J))
      CALL SECOND(YH,VMH,M,RP(I,J),VV(I,J))
      CALL SECOND (YM,SRH,M,RP(I,J),PT(I,J))
      IF (J.EQ.1) PT(I,J)=SR(1,1)
    8 CALL ROTATE (XH(I,J),RP(I,J),-SIN1,COS1,1.0)
      SINO=SIN1
    9 COSO=COS1
      WRITE (6,13) XM,GAMA,ALDEG,X1,BDRAG
      WRITE (4,13) XM,GAMA,ALDEG,X1,BDRAG
      DO 10 I=1,NRING
      WRITE (6,12) I
      DO 10 J=1,MERIDS
      XX=RP(J,I)*COS(PSI(J)/RAD)
      Z=RP(J,I)*SIN(PSI(J)/RAD)
      PTS=PT(J,I)
      THS=TH(J,I)
      PSIS=PSI(J)
      STS=SIN(THS/RAD)
      U=STS*COS(PSIS/RAD)
      V=COS(THS/RAD)
      W=STS*SIN(PSIS/RAD)
      CALL ROTATE (XH(J,I),Z,-SINA,COSA,1.0)
      CALL ROTATE (V,W,-SINA,COSA,0.0)
      SQ=U**2+W**2
      PSIS=SIGN(ACOS(U/SQRT(SQ)),W)*RAD
      THS=ACOS(V/SQRT(V**2+SQ))*RAD
      WRITE(6,14) Z,XX,VV(J,I),PTS,THS,PSIS
      WRITE(4,14) Z,XX,VV(J,I),PTS,THS,PSIS
```

```
10    CONTINUE
C
C
11    FORMAT (1H1,5X,17HANGLE OF ATTACK =,F5.2,5X,12HCONE ANGLE =,F12.5,
     15X,12HDATA PLANE =F12.7)
12    FORMAT (9H-RING NO.,I3/6X,1HZ,13X,1HX,12X,4HMACH,10X,6HPT/PTO,8X,5
     1HTHETA,8X,3HPSI)
13    FORMAT(3F10.5,2F10.8)
14    FORMAT (2E15.7,F10.7,E15.7,F12.8,F13.8)
      END
```

```
SUBROUTINE ROTATE (X,Y,SINE,COSINE,D)
TEMP=X-D
X=COSINE*TEMP+SINE*Y+D
Y=-SINE*TEMP+COSINE*Y
RETURN
END
```

```
      SUBROUTINE SECOND (X,Y,N,XP,YP)
C     2ND ORDER LAGRANGE INTERPOLATION SUBROUTINE
      DIMENSION X(1),Y(1)
      M=N-1
      ARROW=XP-X(1)
      NI=N/2+1
      IF ((X(NI)-X(1))*ARROW) 1,1,2
1     IP=2
      GO TO 4
2     DO 3 I=2,M
      IP=I
      IF ((XP-X(I))*ARROW) 4,4,3
3     CONTINUE
4     A=XP-X(IP-1)
      B=XP-X(IP)
      C=XP-X(IP+1)
      D=X(IP-1)-X(IP)
      E=X(IP-1)-X(IP+1)
      G=X(IP)-X(IP+1)
      T1=Y(IP-1)/(D*E)
      T2=-Y(IP)/(D*G)
      T3=Y(IP+1)/(E*G)
      YP=T1*B*C+T2*A*C+T3*A*B
      IF (N.EQ.2) YP=(Y(IP)*A-Y(IP-1)*B)/(A-B)
      RETURN
      END
```

SECTION V

FUNCTIONAL DESCRIPTION OF SUBROUTINES

OVERLAY (BLUNT, 0, 0)

IVALUE	-	Main program. Calls each subordinate program in sequence.
ENTRPG	-	General Lagrangian interpolation.

OVERLAY (BLUNT, 1, 0)

BLUNTS	-	Main program for blunt body solution. Reads in data, sets up initial parameters, and governs course of iteration for the correct shock wave shape.
BODY	-	Determines equation for body, given the set of body points, and determines the coefficients for the equations describing the Mach number variation along the body.
XY	-	Calculates the Cartesian coordinates of a field, body or shock point.
DIFATE	-	Performs a seven point central difference calculation to determine the first and second derivatives of data along a given shock, field or body line.
IDTHER	-	Determines the ideal gas thermodynamic properties of a given point in the flow field.
IMAIN	-	Carries out the inverse blunt body solution based on the parameters specified by BLUNTS

OVERLAY (BLUNT, 2, 0)

IVLS	-	Main program for determining the properties along a line through the blunt body flow field in order to start the rotationally symmetric method of characteristics solution
DETER	-	Interpolates for all properties at a given point.
BBOUT	-	Prints out the final correct blunt body solution.

OVERLAY (BLUNT, 3, 0)

MOCS	-	Main program for rotationally symmetric method of characteristics solution.
BODI	-	Carries out body point solution.

OPER8	-	Organizes the field and body point solutions and determines when the calculations have been carried sufficiently far downstream.
FIELD	-	Carries out field point solution.
VMACH	-	Calculates Mach number from static and total pressures.
ENTROP	-	Keeps track of the relationship between total pressure and stream function, and, given one of these, determines the other.
TSON	-	Calculates the shock wave angle necessary to develop a downstream Mach number of unity.
DIF	-	Performs linear interpolation along characteristics.
CHANGE	-	Promotes and demotes characteristics during the calculation.
INPUT	-	Reads the input data and makes the necessary dimensional changes.
SHAPE	-	Generates the values of the coefficients defining the body and the step sizes required.
DROP	-	Checks the geometric relationship between points on the initial value line and just downstream of the bow shock and drops any unnecessary points.
OUTPUT	-	Arranges and prints the program output.
WAVE	-	Solves the Rankine-Hugoniot relations to determine the properties downstream of a shock wave.
SHOCK	-	Carries out shock point solution.
MESH	-	Controls step size along the bow shock in order to achieve desired step size along the body.
PAGE	-	Keeps track of lines printed and skips to a new page when necessary. Also prints page number in upper right hand corner.
ANGLE	-	Computes Mach angle from Mach number.
VELOC	-	Calculates V/V_{max} from Mach number.

OVERLAY (BLUNT, 4, 0)

IVSES	-	Main program for the interpolation of the initial value surface.
ROTATE	-	Rotates a vector through a specified angle.
SECOND	-	A specialized second order Lagrangian interpolation.

PART 2: THE THREE-DIMENSIONAL METHOD OF CHARACTERISTICS PROGRAM

SECTION I

INTRODUCTION

The Three-Dimensional Method of Characteristics Program calculates inviscid supersonic flows about smooth three-dimensional bodies at angles of attack. Given an initial value surface (IVS), freestream conditions, and a properly described body, the program determines the flow field downstream of the IVS until a user-specified station is reached. The program can also run back-to-back with the Initial Value Surface Program. In this case, the IVS is automatically provided by the Initial Value Surface Program, and only the freestream conditions and the body description need be provided by the user.

The program uses two temporary files TAPE2 and TAPE4, and on options it reads the IVS from TAPE4 and writes the output on TAPE1.

The Three-Dimensional Method of Characteristics Program has computed a variety of flow fields and has proven to be efficient and versatile. Certain limitations of the program are noted here. The local Mach number must be greater than 1.0 everywhere. Hence the freestream must be supersonic, and the configuration must be such that no subsonic region exists at the juncture between the canopy and the fuselage or between the wing leading edge and fuselage. The body must be smooth, without any surface slope discontinuity (i.e., at any point on the body a unique normal exists). Different numbers of data points per ring (up to a maximum of 48 points) may be assigned in different body regions (up to 20 regions).

SECTION II

BODY DESCRIPTION

Every configuration has a number of generating lines, such as the upper profile, the lower profile, the maximum breadth line, or the wing leading edge. In the present body description procedure, each generating line is divided into a number of segments to permit each segment to be described by a conic-section curve. At each cross section of the configuration, simple analytic curves, e.g., the ellipse or cubic, connect any two adjacent generating lines to form the contour of the corss section. The configuration is thus described analytically by simple low-order curves. For a smooth body a unique normal to the surface exists everywhere, and this condition usually requires slope continuity at the junctures between two contour curves or two segments of a generating line.

The fuselage is located in a right-handed coordinate system where the Y-axis is aligned with the fuselage axis; the X-axis is spanwise and the Z-axis is up. All generating lines are represented by a general curve fit of conic sections in several segments. The conic-section curve takes the form

$$\begin{pmatrix} Z \\ X \end{pmatrix} = PY+Q+SG\left(RY^2 + SY + T\right)^{1/2}$$

where $SG = \pm 1$. A straight line is a special case with $R = S = T = 0$. Each curve can be divided into as many segments as necessary to provide adequate body description. Each segment must be continuous with the previous segment and with very few exceptions the slope must be continuous at the junctures to satisfy the requirement of a unique normal to the surface.

A typical cross section of the wing-body configuration is shown in Figure 14. Through each point marked by a dot, a generating line which is a function of Y passes. For input cards, the following Table shows the curve number corresponding to the two projections of each of the generating lines.

Segment	Type
$\overline{C\phi}$	Ellipse
$\overline{UM}$	"
$\overline{GI}$	"
$\overline{HI}$	"
$\overline{LF_f}$	"
$\overline{3\ 4}$	Circle
$\overline{5\ 6}$	"
$\overline{1\ 2}$	Cubic
$\overline{EG}$	Straight Line
$\overline{FH}$	"
$\overline{℄C}$	"
$\overline{℄U}$	"
$\overline{℄L}$	"

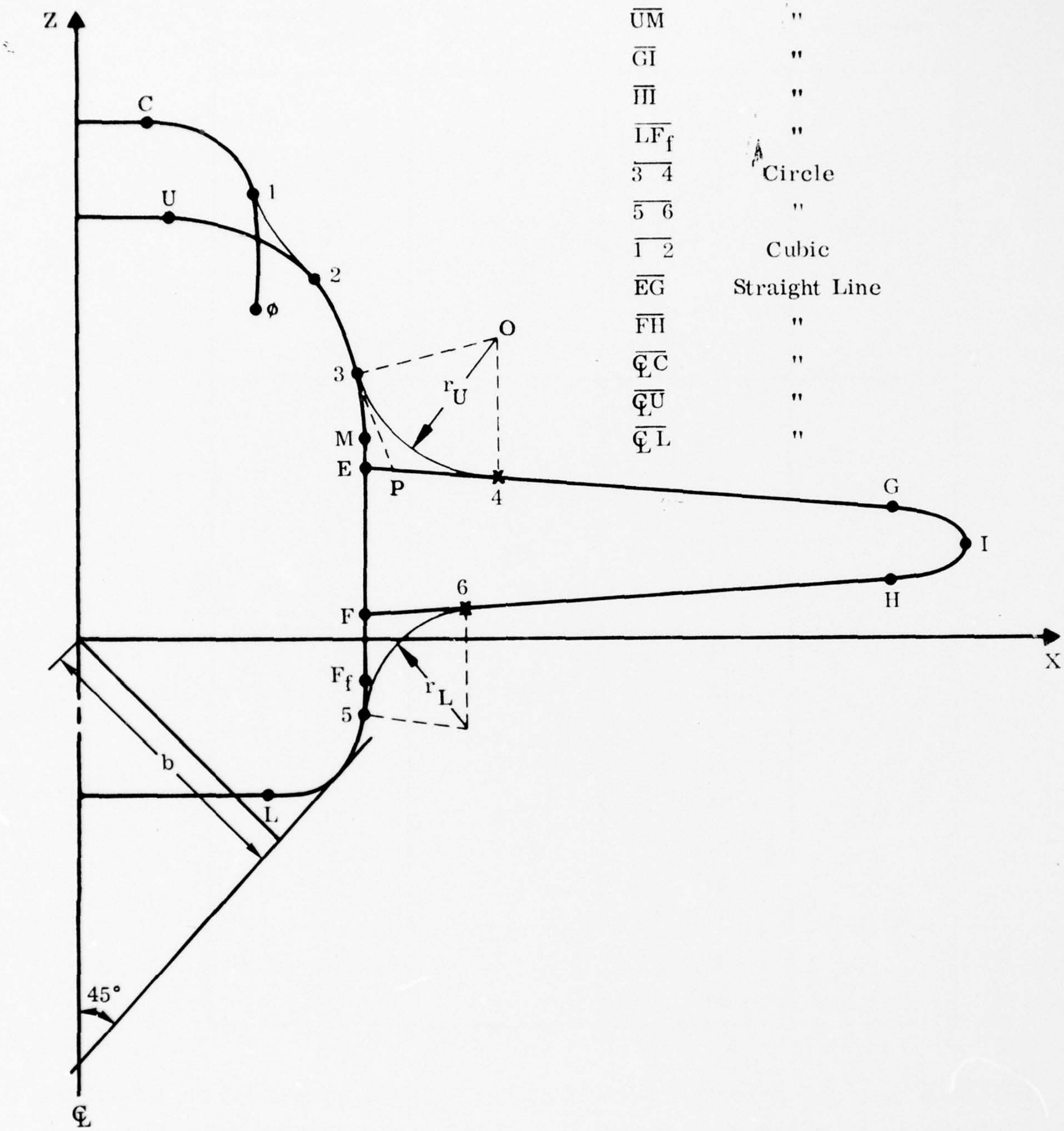


FIGURE 14. SCHEMATIC OF FUSELAGE AND WING CROSS SECTIONS DESCRIPTION

Geometry	Curve No.	Designation Phase I	Designation Phase II
Fuselage	1	Z_U	Z_U
	2	Z_M	Z_M
	3	Z_L	Z_L
	4	X_U	X_U
	5	X_M	X_M
	6	X_L	X_L
	7	b	Z_{F_f}
Canopy	8	Z_1	Z_1
	9	Z_2	Z_2
	10	Z_c	Z_c
	11	Z_ϕ	Z_ϕ
	12	X_ϕ	X_ϕ
	13	X_c	X_c
Wing	14		Z_E
	15		Z_F
	16		Z_G
	17		Z_H
	18		Z_I
	19		X_G
	20		X_H
	21		X_I
	22		Z_3
	23		Z_5

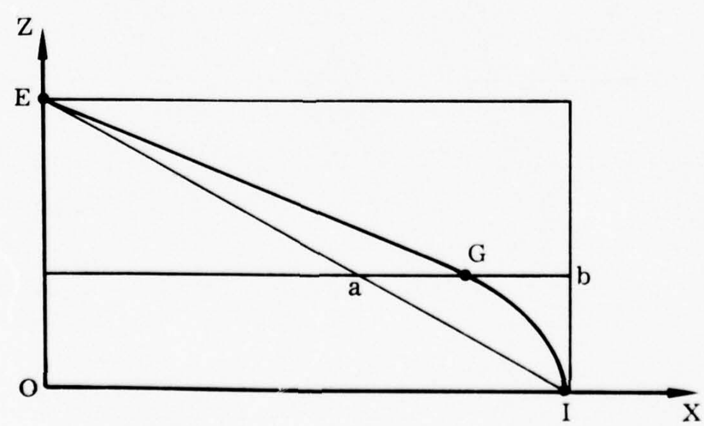
For instance, curve number 3 denotes the projection on the Y-Z plane of the generating line that passes through point L for both Phase I and Phase II designations. Aside from the wing, the only difference between these two phases is in curve 7. While a shape factor b was used in Phase I to alter the lower fuselage, a side flat section was used in Phase II.

As shown in Figure 14, the contour of the cross section begins with a straight line representing the canopy flat. The canopy contour from points C to ϕ is circular but can be elliptic in general. The upper fuselage is represented by an elliptic curve

from points U to M. A straight side flat from points M to F_f joins the upper fuselage to the lower fuselage, which is also represented by an elliptic curve from F_f to L. A straight line from point L to the centerline describes the bottom flat. For the description of the wing, straight lines EG and FH represent the upper and lower surfaces of the wing, respectively. Partial ellipses GI and HI complete the wing description near the leading edge I. The wing-body blending is effected by circular arcs 34 and 56 with radii r_U and r_L, respectively.

In general, the configuration are divided into a number of sections since some of the generating lines can only be described by a composite of conic-section curves. Some of the conic-section curves, however, may remain valid throughout a few sections. To avoid repetition of the input data, the body description subroutine FIGURE reads the data in the following manner (see also Section IV, Input/Output Description). For the first body section, the first card specifies the end of the section and seven conic-section curves. This is followed by 7 cards of body description coefficients, each describing one curve. For a subsequent body section, the first card specifies the end of that particular section and the total number, KT, of conic-section curves that require new values of the coefficients for that section. Only KT cards are then used to supply these new values.

In Figure 14, the projections of lines passing through E, F, G, H and I on the Z-Y plane and X-Y plane are input quantities for defining the wing. In order for the partial ellipses GI and HI to exist, points G and H must be located within a certain range, which depends on the relative position of these five points. When the wing span is very small, it is difficult to input both projections of lines through G and H such that these points are located within acceptable ranges. In such cases, the X coordinates of G and H are calculated inside the program to satisfy the range requirement. As shown in the sketch,



when the projections X_E, Z_E, X_I, Z_I and Z_G are given through input, X_G can be calculated when the ratio $R = Gb/ab$ is specified.

Indeed,

$$\frac{RGb}{X_I - X_E} = \frac{ab}{X_I - X_E} = \frac{Z_G - Z_I}{Z_E - Z_I}$$

and $$X_G = X_I - Gb = X_I + \frac{Z_G - Z_I}{R} \left(\frac{X_E - X_I}{Z_E - Z_I} \right)$$

when fitting the generating lines, a fuselage station is selected ahead of which the projection X_G is not fitted because X_G will be generated by the program according to the above formular with a constant R taken at this fuselage station and input to the program. Since at this station the derivative of X_G with respect to Y can be calculated from the formula, this derivative should be used to fit the projection X_G for fuselage stations immediately behind the selected station.

SECTION III

GRID ARRANGEMENT

Since the geometry of the wing-body configuration is fairly complicated, a procedure was developed to arrange the grid points for efficient calculation. The contour of a wing-body cross section has a number of regions that undergo sharp changes of curvature, where a very fine grid structure is needed. On the other hand, in regions only mild changes of curvature occur, a coarse grid structure suffices.

A sketch of a typical contour is shown in Figure 15, together with the grid points and the labeling of controlling key points. Quantities K_1, K_2, ..., K_8 are specified through input data; for instance, if $K_7 = 20$, it means K_7 is the 20th point counting from the lower profile. Quantities K_9 to K_{12} are obtained by adding to K_7 the differences between K_7 and K_6 to K_3, respectively; for instance, if $K_6 = 17$, then $K_9 = 23$. The quantity K_8 specifies the number of points to be added in regions near the wing-body juncture and the wing leading edge; for instance, $K_8 = 2$ as illustrated. Basically, single grid spacings are assigned from K_1 to K_2, K_3 to K_{5+1}, K_{6-1} to K_{9+1}, K_{10-1} to K_{12}, not counting the additional points specified by K_8. Anywhere else double spacings are assigned. The grid size below the wing leading edge, where K_7 is located, is in general different from that above the leading edge and the difference depends on the value of K_7. Additional points specified by K_8 are added to each single grid space from K_4 to K_5, K_6 to K_9, K_{10} to K_{11}. Care and some experience are needed in assigning the K's. However, the program has the capability to center the series of points from K_4 to K_5 (and hence K_{10} to K_{11}) around the wing-body juncture. The group of points from K_1 to K_2 is designed to cover the fuselage-canopy juncture. All K's are set to zero at fuselage stations before the canopy. Between fuselage-canopy junction and the wing-body juncture only K_1 and K_2 are assigned proper values while all other K's are set to zero.

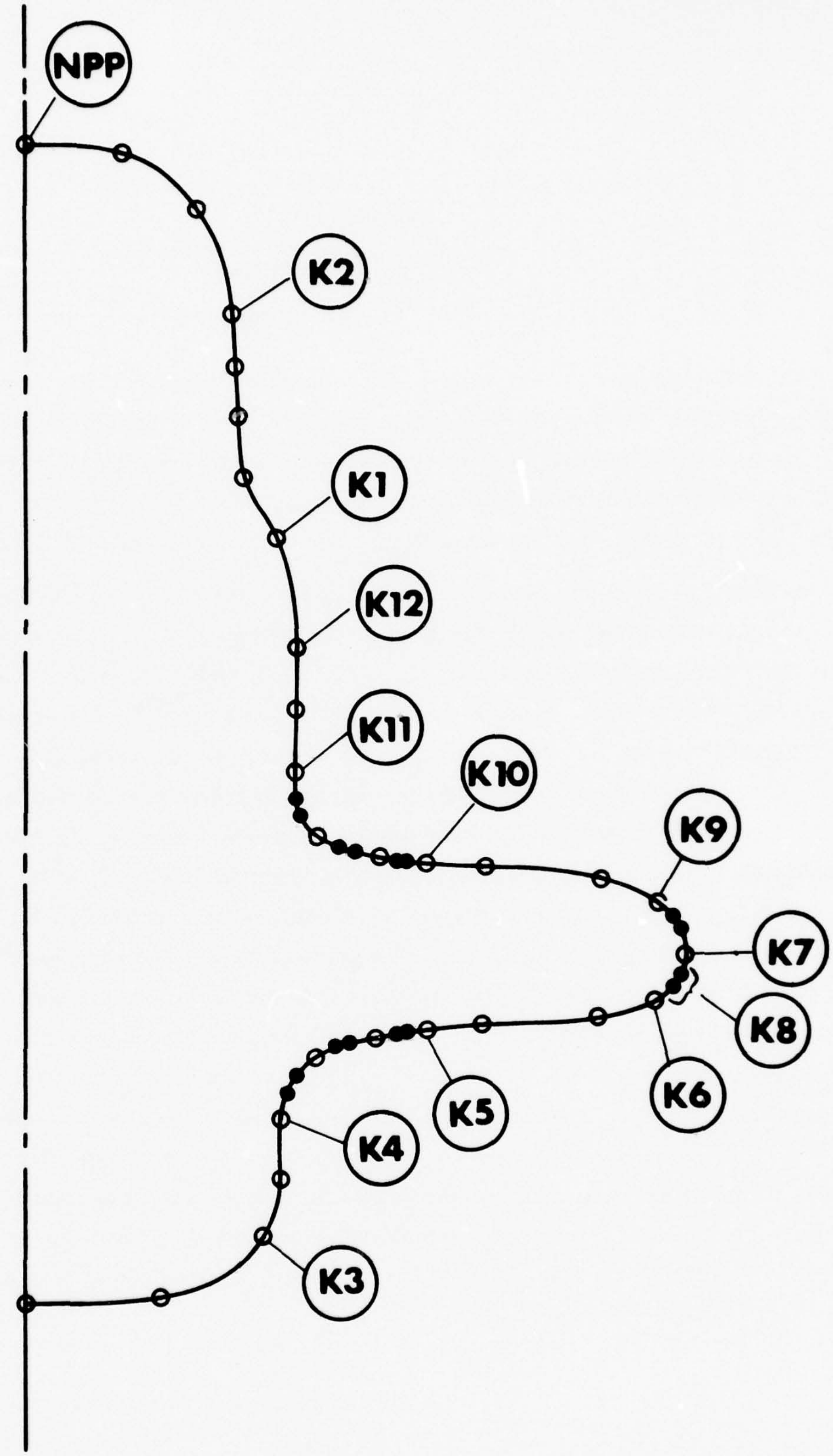


FIGURE 15. GRID STRUCTURE AND CONTROLLING DATA POINTS

SECTION IV

INPUT/OUTPUT DESCRIPTION

1. INPUT DESCRIPTION

The Three-Dimensional Method of Characteristics Program offers some input and output options. The program can run using the IVS data on punched cards or it can run back-to-back with the initial Value Surface Program, which provides the IVS data through a temporary file, TAPE4.

The input data cards required by the Three-Dimensional Method of Characteristics Program are shown in Figure 16. They are described in detail below.

Card No	Variable	Format	Description
①	LIT	18A4	Problem identification
②	DP	F10.5	Convergence criterion for P, 0.1 to 10^{-5}
	DTHETA	F10.5	Convergence criterion for θ;* 0.1 to 10^{-5}
	DPSI	F10.5	Convergence criterion for ψ;* 0.1 to 10^{-5}
	EPSLN	F10.5	Convergence criterion for μ, 0.1 to 10^{-5}
	DYC	F10.5	Initial step size which satisfies the Courant-Friedrichs-Lewy stability condition, i.e., $DYC \cdot \tan\mu < \sqrt{2}\, d$, where μ is the local Mach angle and d is the shortest distance between two data points. If DYC = 0, the program will generate an initial step size.
	YSTOP	F10.5	Computation stops when Y ≥ YSTOP. YSTOP cannot be set equal to 0.
	STEPM	F10.5	Maximum step size (when STEM > 0)
	WINGST†	F10.5	Fuselage station at which the wing begins (110 for the sample case)

*The relationship between the flow angle θ or ψ and the coordinate system is shown in Figure 16a.

†Variables or cards so marked are not needed for the Phase I program.

Card																
1	LIT															
2	DP		DTHETA		DPSI		EPSLN		DYC		YSTOP		STEPM		WINGST†	
3	NSURF	LINE	NPC	NR	IFT	ISTART	KP4	NS	NL	NP	KPR(1)	KPR(2)	KPR(3)			
4	YSET (I)		N0 (9I-8)	N0 (9I-7)	N0 (9I-6)	N0 (9I-5)	N0 (9I-4)	N0 (9I-3)	N0 (9I-2)	N0 (9I-1)	N0 (9I)					
	● Additional data as indicated †															
5	FMSTR		GF (1)		ALFA		Y0									
6	YA		ZA		SCALE		WINGA		DBLNL†							
7†	NCQ															
8†	J	CQ (1, J)		CQ (2, J)												
9†	IC	CQ (3, J)		CQ (4, J)		CQ (5, J)		CQ (6, J)		CQ (7, J)		CQ				
	● Additional data as indicated															
	● See remark on cards 10, 11, 12, 13															
10	ISURF	LINE	ILL	KURE	MESS	INCRES	NPC	K1 (1)	K1 (2)	K1 (3)	K1 (4)	K1 (5)	K1 (6)	K1 (7)	K1 (8)	
11	SINE		COSINE		S2		C2									
12	DYC		STEP		Y0		DRAG									
13	ALIFT		TORQUE													
	● See remark on card 14															
14	A (J, 3, I)		A (J, 1, I)		A (J, 7, I)		A (J, 8, I)		THETT (J, I)		PSIT (J, I)					
	● Additional data as indicated															
15	YMAX	N	MOD†	RATIO†												
16	L	P (L)	Q (L)	R (L)	S (L)	T (L)	SG									
	● Additional data as indicated															
	● See remark on card 17															
17	YMAX	KT	MOD†	RATIO†												
18	M	P (M)	Q (M)	R (M)	S (M)	T (M)	SG									
	● Additional data as indicated															

†Variables or cards so marked are not needed for the Phase I program.

FIGURE 16. INPUT DATA FOR THREE-DIMENSIONAL METHOD OF CHARACTERISTICS PROGRAM

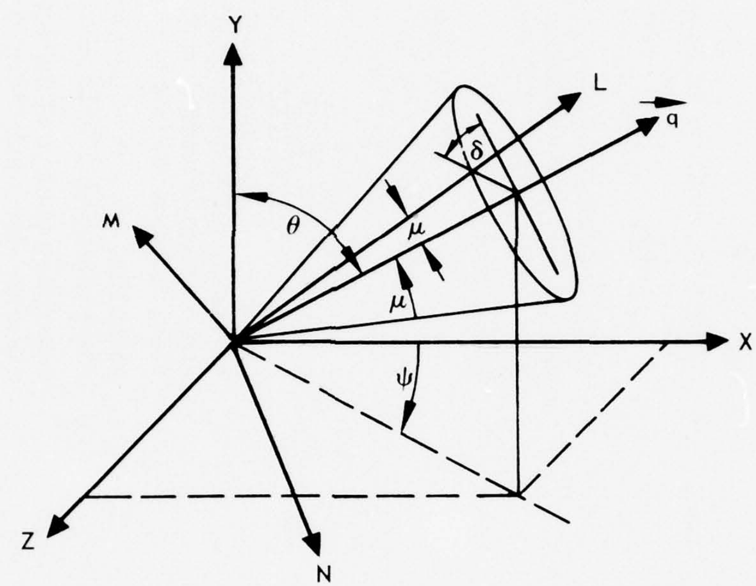


FIGURE 16a. RELATION BETWEEN COORDINATE SYSTEMS

Card No	Variable	Format	Description
	NSURF	I5	Total number of surfaces desired, counting from IVS, even for a restart from a data surface
	LINE	I5	Number of data rings per surface. LINE is updated by data from card 7 for a restart from cards
	NPC	I5	Number of data points per ring. NPC is updated by data from card 7 for a restart from cards
	NR	I5	Number of regions having different NPC. $1 \leq NR \leq 20$
	IFT	I5	Control for reading IVS data. If IFT = 0, IVS data is on cards; = 1, IVS data is on TAPE4. Also, if IFT = K < 0, then number of rings per data surface is increased by $\|K\|$ during restart from TAPE1. $\|K\| \leq 2$ is recommended.
	ISTART	I5	Restart control* If ISTART < 0, restart from cards; = 0, initial run; = I > 0, restart from Ith data surface on TAPE1
3	KP4	I5	Output control. If KP4 = K > 0, every data surface is written on TAPE1, with every Kth surface printed out. If KP4 = K ≤ 0, every data surface is printed out. If K = 0, every 10th data surface is written on TAPE1. If K < 0, every Kth data surface is punched out on cards.
	NS	I5	Identifying number of the surface on which print level KPRINT (1) will apply
	NL	I5	Identifying number of the ring on data surface NS along which print level KPRINT (2) will apply
	NP	I5	Identifying number of the data point along ring NL on data surface NS at which print level KPRINT (3) will apply
	KPRINT (1)	I5	Print level control,** usually 0. If KPRINT (1) = 0, normal print; = 1, moderate default print; = 2, extensive default print
	KPRINT (2)	I5	Print level control, usually 0 or 1
	KPRINT (3)	I5	Print level control. See Section V

*For restart, insure ISTART ≠ 0, or data on TAPE1 will be ruined
**See Section V for warning on print level

Card No	Variable	Format	Description
④	YSET(I)	F10.5	For all Y ≤ YSET (I), the number of points per ring to be calculated is specified by the variable NPSET(9I-8), and the grid size is specified by NPSET(9I-7) . . . NPSET(9I) (See Section III where K_1 denotes NPSET(9I-7) . . . K_8 denotes NPSET(9I))*
	NPSET(9I-8)	I5	
	NPSET(9I-7)	I5	
	⋮	⋮	
	NPSET(9I)†	I5	

● There now follow cards, I = 2, NR

Card No	Variable	Format	Description
⑤	FMSTR	F10.5	Freestream Mach number
	GF (1)	F10.5	Ratio of specific heats
	ALFA	F10.5	Angle of attack, in degrees
	Y0	F10.5	Y-Coordinate of initial value surface. Needed only for an initial run using IVS data from cards. May be left blank for all other cases.
⑥	YA	F10.5	Y-coordinate of center of spherical nose
	ZA	F10.5	Z-coordinate of center of spherical nose
	SCALE	F10.5	Scale factor (radius of spherical nose)
	WINGA	F10.5	Wing area, Sq. Ft. (reference area for C_{DW})
	DBLNL†	F10.5	Distance between points 5 and F in Figure 14 (0.5 for sample case)
⑦†	NCQ	I5	Number of sections needed to fit the blending curve (#22 for the sample case) by conic sections
⑧†	J	I2	Section number
	CQ(1, J)	E15.7	Fuselage station representing lower limit of the section
	CQ(2, J)	E15.7	Fuselage station representing higher limit of the section

*Note that for the Phase I program only K_1 and K_2 are used.

†Variables or cards so marked are not needed for the Phase I program.

Card No	Variable	Format	Description
9†	IC	I2	Curve number for blending (#22 for the sample case)
	CQ(3, J)	E15.7	Conic coefficient P(J)
	⋮	⋮	⋮
	CQ(7, J)	E15.7	Conic coefficient T(J)
	CQ(8, J)	F2	Conic coefficient SG(J)

- There now follow sets of cards, J = 2, NCQ, each set describing a section.
- If ISTART ≥ 0, omit cards 10, 11, 12, 13
- Note: Cards 10, 11, 12, 13 are required only for a restart using data on punched cards. They are generated by the 3DMoC Program when the flow field data is punched out through the KP4 option. Value of the restart parameters are also part of the printout for each data surface. The variable INCRES may be set at a different value as explained below.

Card No	Variable	Format	Description
10	ISURF	I5	Restart parameters
	LINE	I5	
	ILL	I5	
	KURE	I5	
	MESS	I5	
	INCRES	I5	If LINE is the number of rings per data surface during a restart from cards, then the new number will be LINE + INCRES. INCRES ≤ 2 is recommended.
	NPC	I5	Restart parameters*
	KI(1)	I5	
	⋮	⋮	
	KI(8)†	I5	
11	SINE	E15.8	Restart parameters
	COSINE	E15.8	
	S2	E15.8	
	C2	E15.8	

*When making a restart after an initial run, be sure to remove IVS data cards, if they were present for the initial run.

Card No	Variable	Format	Description
12	DYC	E15.8	Restart Parameters
	STEP	E15.8	
	Y0	E15.8	
	DRAG	E15.8	
13	ALIFT	E15.8	Restart Parameters
	TORQUE	E15.8	
	● GO TO card 15 unless		ISTART = 0 and IFT = 0, or ISTART < 0
14	A(J, 3, I)	E15.8	Z-coordinate of data point
	A(J, 1, I)	E15.8	X-coordinate of data point
	A(J, 7, I)	F10.7	Mach number at data point
	A(J, 8, I)	E15.8	Total pressure ratio, P_t/P_{t_∞}
	THETT(J, I)	F12.8	Flow angle θ
	PSIT(J, I)	F13.8	Flow angle Ψ

- There now follow cards, J = 2, NPC which describe a ring on the data surface
- There now follow sets of cards, I = 2, LINE with each set describing a ring on the data surface

Card No	Variable	Format	Description
15	YMAX	F10.5	Y-coordinate of the end of the first body section described by the coefficients to follow.
	N	I5	Number of body description curves for the body sections. Must be set equal to 7.
	MOD†	I5	= 1, fuselage only; = 2, fuselage - canopy; = 3, fuselage - wing; = 4 fuselage - canopy-wing
	RATIO†	F10.5	Ratio for partial ellipse (see Section II) If Ratio > 0, curves 19 and 20 are generated by the program.**

**For the sample case, RATIO > 0 between F.S. 110 and F.S. 220

Card No	Variable	Format	Description
16	L	I2	Identifying number of the body description curve
	P(L)	E15.8	Body description coefficients (See Section II)
	Q(L)	E15.8	
	R(L)	E15.8	
	S(L)	E15.8	
	T(L)	E15.8	
	SG(L)	F2.0	

- There now follow cards, L = 2, 6
- If the body is described by only one section, no further data is required

Card No	Variable	Format	Description
17	YMAX	F10.5	Y-coordinate of the end of the next body section described by the coefficients to follow
	KT	I5	Total number of body description curves that need new values of P, Q, R, S, T, SG for this body section
	MOD†	I5	See card 15
	RATIO†	F10.5	See card 15
18	M	I2	Identifying number of the body description curve which requires new coefficients
	P(M)	E15.8	Body description coefficients
	Q(M)	E15.8	
	R(M)	E15.8	
	S(M)	E15.8	
	T(M)	E15.8	
	SG(M)	F2.0	

- There now follow a total of KT-1 cards, giving the new coefficients for this body section
- There now follow sets of cards, starting with card type 17, each set describing the subsequent body sections to the end of the configuration

2. OUTPUT DESCRIPTION

A number of output options are available through the control index KP4, which is part of the input data. If KP4 = K > 0 , every data surface is written on TAPE1 with every Kth data surface printed out for sampling. If KP4 = K < 0, every data surface is printed out, while retaining certain data surfaces for a restart (see also Section V, Operational Aspects). If K = 0, every 10th data surface is written on TAPE1. If K is negative, then every Kth surface is punched on cards.

When any output is written on TAPE1, the user should either use a physical tape as TAPE1, through a request control card, or copy TAPE1 to a physical tape as many times as there are data surfaces through a COPYCF control card before termination. Otherwise, information on TAPE1, which is a scratch disc file, will be lost upon termination of the run. A restart at a later time can be made from the physical tape. The data surfaces on the tape can be printed out by copying them to OUTPUT through a control card COPYCF (NAME, OUTPUT, N), where NAME is the tape designation and N is the number of data surfaces to be printed out. The tape can also be taken to a peripheral processor for printing of the data surfaces.

Figure 17 shows a normal printout for a data surface. X, Y, Z, are the coordinates of a data point (see Figure 15 for the coordinate system). P is the static pressure and PT is the total pressure, with the subscript 0 referring to freestream quantities. The angles OUTWASH and UPWASH, in degrees, are included since they are more meaningful than θ and ψ. These angles are related to the direction cosines u, v, w of the velocity vector by: OUTWASH = atan(u/v), UPWASH = atan(w/v), θ = acos(v), and ψ = atan(w/u).

Figure 18 shows a data surface where the CFL stability condition was violated at certain data points. When the velocity vector is aligned with the Y-axis, certain flow angles are undefined. When this condition is detected by the program, the coordinate system is automatically rotated and the particular data surface is recomputed. An example of such occurrences is shown in Figure 19.

When the normal output is written on TAPE1, every Kth surface is printed out for sampling. An example of this is shown in Figure 20, where the 5th data surface is printed out completely, while the upper and lower centerline pressure coefficients on the body surface are printed out for data surfaces 6 through 9.

DATA SURFACE NO 50

***TO USE THIS AS AN INITIAL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS

SURFACE NO 50, STATION Y = .69602455E+01

BODY POINTS

Z	X	MACH NO.	PT/PTO	THETA, DEG	PSI,DEG	CP	P/PO	P/PTO	OUTWASH	UPWASH
-.26323191E+01	-.20385704E-11	1.6777867	.67039950E-02	14.99999998	-90.00000000	.33202E+00	.62428E+02	.12816E-02	0.00	-15.00
-.24930217E+01	.84483781E+00	1.7119138	.67039950E-02	15.34021658	-58.90000772	.31581E+00	.59428E+02	.12200E-02	8.06	-13.22
-.20918693E+01	.15977245E+01	1.8123940	.67039950E-02	16.19329874	-29.94616641	.27245E+00	.51407E+02	.10554E-02	14.12	-8.25
-.14762228E+01	.21793025E+01	1.9760670	.67039950E-02	17.17117257	-4.23829123	.21421E+00	.40632E+02	.83416E-03	17.13	-1.31
-.71647510E+00	.25328311E+01	2.1886184	.67039950E-02	17.88033751	18.05198180	.15704E+00	.30054E+02	.61700E-03	17.05	5.71
.10587877E+00	.26300771E+01	2.4306471	.67039950E-02	18.12135956	37.34813306	.11077E+00	.21493E+02	.44125E-03	14.58	11.23
.91089550E+00	.24696906E+01	2.6926111	.67039950E-02	17.93319395	54.35713544	.76390E-01	.15133E+02	.31068E-03	10.66	14.74
.16181831E+01	.20761618E+01	2.9180800	.67039950E-02	17.18787480	67.90816241	.55742E-01	.11313E+02	.23225E-03	6.64	15.99
.21667872E+01	.14945004E+01	3.0737217	.67039950E-02	16.17658692	77.93828499	.44925E-01	.93117E+01	.19117E-03	3.47	15.84
.25136693E+01	.78123145E+00	3.1527414	.67039950E-02	15.32772321	84.89024934	.40276E-01	.84516E+01	.17351E-03	1.40	15.27
.26323191E+01	-.95484356E-12	3.1740096	.67039950E-02	14.99999998	90.00000000	.39110E-01	.82358E+01	.16908E-03	0.00	15.00

FIELD POINTS

Z	X	MACH NO.	PT/PTO	THETA, DEG	PSI,DEG	CP	P/PO	P/PTO	OUTWASH	UPWASH
-.27976743E+01	-.30708077E-11	2.6656985	.27292557E-01	13.31310132	-90.00000000	.33960E+00	.63829E+02	.13104E-02	0.00	-13.31
-.26524812E+01	.90346439E+00	2.6963417	.27043901E-01	13.51892781	-61.23774330	.32294E+00	.60748E+02	.12472E-02	6.60	-11.90
-.22346296E+01	.17137388E+01	2.7888730	.26363194E-01	14.04846027	-34.16076513	.27832E+00	.52492E+02	.10776E-02	11.70	-8.00
-.15948933E+01	.23547420E+01	2.9377430	.25123766E-01	14.71087442	-9.76328100	.21812E+00	.41355E+02	.84901E-03	14.51	-2.55
-.79941041E+00	.27727044E+01	3.1232295	.23336648E-01	15.34037100	11.69186140	.15945E+00	.30500E+02	.62616E-03	15.04	3.18
.75389643E-01	.29387271E+01	3.3255777	.21341877E-01	15.78929729	30.20675453	.11277E+00	.21864E+02	.44887E-03	13.73	8.10
.10540670E+01	.28449322E+01	3.5724237	.19633626E-01	16.04806569	47.64526664	.76288E-01	.15114E+02	.31029E-03	10.97	12.00
.19396953E+01	.24431095E+01	3.7451265	.17649843E-01	16.06534175	61.98431124	.55156E-01	.11204E+02	.23003E-03	7.70	14.26
.26457122E+01	.17857145E+01	3.8498956	.16066152E-01	15.84616070	73.40760919	.43773E-01	.90985E+01	.18679E-03	4.63	15.22
.31012817E+01	.94243880E+00	3.8991642	.15053060E-01	15.58485241	82.40581379	.38294E-01	.80848E+01	.16598E-03	2.11	15.45
.32587489E+01	-.18649713E-11	3.9115300	.14693739E-01	15.47540440	90.00000000	.36690E-01	.77881E+01	.15989E-03	0.00	15.48

FIELD POINTS

Z	X	MACH NO.	PT/PTO	THETA, DEG	PSI,DEG	CP	P/PO	P/PTO	OUTWASH	UPWASH
-.29665304E+01	-.30161765E-11	3.2299571	.54116050E-01	11.91968348	-90.00000000	.33045E+00	.62137E+02	.12757E-02	0.00	-11.92
-.28122352E+01	.97196388E+00	3.2810111	.54748474E-01	12.13770859	-60.35033663	.31423E+00	.59136E+02	.12141E-02	6.07	-10.59
-.23697403E+01	.18450779E+01	3.4319847	.56536971E-01	12.66522871	-33.07388640	.27096E+00	.51131E+02	.10497E-02	10.66	-6.99
-.16971259E+01	.25444420E+01	3.6658618	.58368023E-01	13.27107181	-9.18257228	.21324E+00	.40452E+02	.83048E-03	13.11	-2.16
-.86170126E+00	.30200117E+01	3.9486194	.59265227E-01	13.84063114	11.11311646	.15780E+00	.30195E+02	.61991E-03	13.59	2.72
.63255148E-01	.32468202E+01	4.2401023	.58524033E-01	14.31354213	28.00018866	.11376E+00	.22046E+02	.45261E-03	12.70	6.83
.12081033E+01	.32170727E+01	4.5561987	.55517134E-01	14.91969234	45.34500773	.77408E-01	.15321E+02	.31454E-03	10.61	10.73
.22664129E+01	.28074301E+01	4.7705603	.50998056E-01	15.45497728	59.46456577	.56754E-01	.11500E+02	.23610E-03	8.00	13.39
.31260502E+01	.20764533E+01	4.9044262	.47121327E-01	15.74299546	71.11017791	.45400E-01	.93995E+01	.19297E-03	5.21	14.93
.36889106E+01	.11040547E+01	4.9667372	.44204069E-01	15.84234371	81.03921897	.39650E-01	.83357E+01	.17113E-03	2.53	15.66
.38848609E+01	-.22887734E-11	4.9836407	.43134643E-01	15.85947158	90.00000000	.37899E-01	.80117E+01	.16448E-03	0.00	15.86

FIELD POINTS

Z	X	MACH NO.	PT/PTO	THETA, DEG	PSI,DEG	CP	P/PO	P/PTO	OUTWASH	UPWASH
-.31351451E+01	-.30659467E-11	3.4733950	.68179800E-01	10.47699753	-90.00000000	.31228E+00	.58776E+02	.12067E-02	0.00	-10.48
-.29713223E+01	.10425048E+01	3.5470307	.70598330E-01	10.78636306	-57.80054023	.29694E+00	.55938E+02	.11484E-02	5.80	-9.16
-.25023870E+01	.19809951E+01	3.7633794	.77823781E-01	11.51956714	-29.1123?850	.25633E+00	.48424E+02	.99413E-03	10.10	-5.66
-.17933179E+01	.27409980E+01	4.1003025	.88867509E-01	12.35586870	-5.02107111	.20335E+00	.38621E+02	.79289E-03	12.31	-1.10
-.91347484E+00	.32749428E+01	4.4957342	.10116011E+00	13.17438674	14.38342325	.15454E+00	.29593E+02	.60753E-03	12.78	3.33
.64156103E-01	.35609046E+01	4.8916615	.11230892E+00	13.87056906	29.43056499	.11709E+00	.22664E+02	.46528E-03	12.14	6.92
.13713641E+01	.35955241E+01	5.2997805	.11960076E+00	14.90520563	45.10801612	.85693E-01	.16854E+02	.34601E-03	10.64	10.68
.26015709E+01	.31784251E+01	5.5966917	.12265760E+00	15.93036388	57.97844309	.67537E-01	.13495E+02	.27705E-03	8.61	13.60
.36159074E+01	.23736922E+01	5.7875031	.12294628E+00	16.72504138	69.18231679	.57311E-01	.11603E+02	.23821E-03	6.10	15.69
.42882936E+01	.12700193E+01	5.8910988	.12182711E+00	17.23176789	79.71300680	.51874E-01	.10597E+02	.21756E-03	3.17	16.97
.45239423E+01	-.23908963E-11	5.9236406	.12123839E+00	17.40828270	90.00000000	.50170E-01	.10282E+02	.21109E-03	0.00	17.41

SHOCK POINTS

Z	X	MACH NO.	PT/PTO	THETA, DEG	PSI,DEG	CP	P/PO	P/PTO	OUTWASH	UPWASH
-.33755736E+01	-.30335538E-11	3.6197884	.71476644E-01	8.39261285	-90.00000000	.27652E+00	.52160E+02	.10708E-02	0.00	-8.39
-.32009045E+01	.11265813E+01	3.7110667	.75753007E-01	8.90781522	-53.85709166	.26447E+00	.49930E+02	.10251E-02	5.28	-7.21
-.27003179E+01	.21450391E+01	3.9770099	.89203236E-01	10.10834147	-23.80404442	.23297E+00	.44102E+02	.90541E-03	9.26	-4.12
-.19431837E+01	.29804246E+01	4.3874689	.11294150E+00	11.49100404	-.44116181	.19310E+00	.36726E+02	.75397E-03	11.49	-.09
-.99957110E+00	.35848801E+01	4.8511827	.14423618E+00	12.90737422	17.34666876	.15783E+00	.30200E+02	.62000E-03	12.34	3.91
.55381788E-01	.39350575E+01	5.2953026	.17872129E+00	14.13077835	30.48329774	.13124E+00	.25282E+02	.51903E-03	12.24	7.28
.15389353E+01	.40362166E+01	5.7528929	.21882447E+00	15.86885527	44.64861526	.10933E+00	.21228E+02	.43580E-03	11.43	11.30
.29624946E+01	.36054105E+01	6.0922807	.25147079E+00	17.49489797	57.29123861	.95857E-01	.18735E+02	.38462E-03	9.67	14.85
.41491131E+01	.27127502E+01	6.3029156	.27292136E+00	18.71557603	68.47196915	.88468E-01	.17368E+02	.35656E-03	7.09	17.49
.49421380E+01	.14580109E+01	6.4300267	.28628909E+00	19.47717631	79.28847527	.84326E-01	.16601E+02	.34082E-03	3.76	19.16
.52211780E+01	-.22729670E-11	6.4719569	.29076667E+00	19.73873885	90.00000000	.83008E-01	.16357E+02	.33582E-03	0.00	19.74

TO USE AS IVS, PUNCH THE 3 CARDS, PLACE BEFORE BODY PT CARDS

```
  50    5    0    0    0    0    0    0   11
 0.             .10000000E+01 0.             .10000000E+01
 .19505687E+00  .50000000E-01  .69602455E+01
```

FIGURE 17. NORMAL DATA SURFACE

DATA SURFACE NO 47

***TO USE THIS AS AN INITIAL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS

SURFACE NO 47, STATION Y = .64024979E+01

BODY POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.24828711E+01	-.20385704E-11	1.6974006	.67039950E-02	14.99999998	-90.00000000	.32261E+00	.60686E+02	.12459E-02	0.00	-15.00
-.23472154E+01	.80935275E+00	1.7327769	.67039950E-02	15.35006741	-58.42099948	.30628E+00	.57665E+02	.11839E-02	8.18	-13.16
-.19580005E+01	.15266059E+01	1.8362326	.67039950E-02	16.21600179	-29.17418670	.26307E+00	.49671E+02	.10197E-02	14.25	-8.07
-.13647476E+01	.20740713E+01	2.0023023	.67039950E-02	17.18056230	-3.41336968	.20613E+00	.39136E+02	.80345E-03	17.15	-1.05
-.63910506E+00	.23991367E+01	2.2134315	.67039950E-02	17.84895448	18.76855390	.15148E+00	.29026E+02	.59589E-03	16.96	5.92
.13903661E+00	.24788812E+01	2.4490689	.67039950E-02	18.04852226	37.89843246	.10789E+00	.20940E+02	.43011E-03	14.42	11.32
.89418079E+00	.23161535E+01	2.6982275	.67039950E-02	17.82957909	54.69630300	.75789E-01	.15022E+02	.30840E-03	10.53	14.71
.15517372E+01	.19382250E+01	2.9066942	.67039950E-02	17.10118700	68.11550318	.56632E-01	.11478E+02	.23563E-03	6.54	15.93
.20573989E+01	.13896798E+01	3.0472019	.67039950E-02	16.13565204	78.12381887	.46604E-01	.96223E+01	.19754E-03	3.41	15.81
.23748018E+01	.72441631E+00	3.1161477	.67039950E-02	15.31914894	85.03319850	.42366E-01	.88381E+01	.18145E-03	1.36	15.26
.24828711E+01	-.12782267E-11	3.1341602	.67039950E-02	14.99999998	90.00000000	.41324E-01	.86454E+01	.17749E-03	0.00	15.00

BASE POINT 4 VIOLATES CFL .70226

FIELD POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.26400132E+01	-.30111226E-11	2.5875106	.23911452E-01	13.26371613	-90.00000000	.32989E+00	.62034E+02	.12736E-02	0.00	-13.26
-.24999925E+01	.86298769E+00	2.6187700	.23690485E-01	13.48844578	-60.57985591	.31325E+00	.58956E+02	.12103E-02	6.72	-11.80
-.20979703E+01	.16344654E+01	2.7124572	.23088619E-01	14.06340831	-33.03622715	.26906E+00	.50779E+02	.10425E-02	11.88	-7.78
-.14849242E+01	.22407143E+01	2.8620830	.22049761E-01	14.77554676	-8.43346743	.21057E+00	.39958E+02	.82033E-03	14.62	-2.22
-.72619610E+00	.26305927E+01	3.0445390	.20578100E-01	15.43905604	13.01129891	.15477E+00	.29634E+02	.60837E-03	15.06	3.56
.10295106E+00	.27794202E+01	3.2397285	.18959378E-01	15.89371207	31.36373017	.11049E+00	.21515E+02	.44171E-03	13.47	8.43
.10241345E+01	.26807015E+01	3.4766576	.17695570E-01	16.13595393	48.55356021	.76738E-01	.15198E+02	.31200E-03	10.84	12.24
.18520927E+01	.22936112E+01	3.6432330	.16166468E-01	16.11901111	62.65846552	.56765E-01	.11491E+02	.23591E-03	7.66	14.40
.25081436E+01	.16716112E+01	3.7476611	.14933445E-01	15.86869620	73.84678621	.45692E-01	.94531E+01	.19408E-03	4.52	15.27
.29294808E+01	.88060236E+00	3.8000842	.14149829E-01	15.58409797	82.61736639	.40312E-01	.84581E+01	.17364E-03	2.05	15.46
.30747430E+01	-.21828434E-11	3.8143873	.13871183E-01	15.46520429	90.00000000	.38719E-01	.81635E+01	.16760E-03	0.00	15.47

FIELD POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.28021286E+01	-.30296590E-11	3.1747894	.49358509E-01	11.84594392	-90.00000000	.32263E+00	.60579E+02	.12437E-02	0.00	-11.85
-.26542043E+01	.92747647E+00	3.2240174	.49779010E-01	12.06211318	-60.16296564	.30575E+00	.57567E+02	.11818E-02	6.07	-10.50
-.22307021E+01	.17590280E+01	3.3687995	.50962622E-01	12.59373133	-32.70193083	.26278E+00	.49618E+02	.10186E-02	10.65	-6.88
-.15885172E+01	.24226459E+01	3.5899365	.51984252E-01	13.22931604	-8.68341094	.20645E+00	.39233E+02	.80545E-03	13.08	-2.03
-.79253192E+00	.28697980E+01	3.8535595	.52229181E-01	13.45419087	11.66586173	.15384E+00	.29461E+02	.60484E-03	13.58	2.85
.85348625E-01	.30792678E+01	4.1199994	.51137351E-01	14.39000576	28.58362890	.11230E+00	.21776E+02	.44707E-03	12.70	7.00
.11656922E+01	.30418888E+01	4.4075313	.48431157E-01	15.03251952	45.85391592	.78052E-01	.15441E+02	.31699E-03	10.59	10.91
.21590426E+01	.26459277E+01	4.6048467	.44733297E-01	15.55556504	59.90252975	.58293E-01	.11785E+02	.24194E-03	7.95	13.54
.29619787E+01	.19520712E+01	4.7306801	.41607129E-01	15.81297638	71.44162585	.47222E-01	.97366E+01	.19989E-03	5.15	15.03
.34857033E+01	.10362417E+01	4.7914241	.39265847E-01	15.88532346	81.21464480	.41540E-01	.86853E+01	.17831E-03	2.49	15.71
.36676462E+01	-.26286613E-11	4.8086705	.38411218E-01	15.89205878	90.00000000	.39794E-01	.83623E+01	.17168E-03	0.00	15.89

FIELD POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.29643692E+01	-.30531226E-11	3.4676731	.66153414E-01	10.38914212	-90.00000000	.30488E+00	.57406E+02	.11785E-02	0.00	-10.39
-.28074897E+01	.99632412E+00	3.5387789	.68223333E-01	10.69535899	-57.61186436	.28953E+00	.54566E+02	.11202E-02	5.74	-9.06
-.23592482E+01	.18918605E+01	3.7462599	.74353523E-01	11.42895616	-28.80539119	.24940E+00	.47142E+02	.96782E-03	10.05	-5.56
-.16827858E+01	.26153639E+01	4.0626961	.83371782E-01	12.29412941	-4.71762280	.19823E+00	.37675E+02	.77347E-03	12.25	-1.03
-.84401730E+00	.31198113E+01	4.4283853	.93201451E-01	13.15945414	14.54956285	.15195E+00	.29112E+02	.59767E-03	12.75	3.36
.85120067E-01	.33870576E+01	4.7892901	.10182513E+00	13.91496331	29.50040859	.11657E+00	.22567E+02	.46331E-03	12.17	6.96
.13200957E+01	.34111311E+01	5.1534062	.10682459E+00	15.01268720	45.12096764	.86903E-01	.17078E+02	.35061E-03	10.72	10.76
.24778846E+01	.30067362E+01	5.4241190	.10911270E+00	16.06062072	58.00600394	.69421E-01	.13844E+02	.28421E-03	8.67	13.72
.34292431E+01	.22408891E+01	5.5996477	.10931398E+00	16.85557265	69.22673671	.59446E-01	.11998E+02	.24632E-03	6.13	15.82
.40583499E+01	.11974472E+01	5.6948031	.10836608E+00	17.35785299	79.74702311	.54119E-01	.11013E+02	.22609E-03	3.18	17.10
.42785396E+01	-.27498636E-11	5.7247277	.10787998E+00	17.53171628	90.00000000	.52447E-01	.10703E+02	.21974E-03	0.00	17.53

SHOCK POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.32198064E+01	-.30335538E-11	3.6656068	.73601792E-01	8.17885552	-90.00000000	.27039E+00	.51025E+02	.10475E-02	0.00	-8.18
-.30517155E+01	.10849104E+01	3.7549150	.77868495E-01	8.71986397	-53.45597205	.25892E+00	.48903E+02	.10040E-02	5.22	-7.02
-.25707418E+01	.20647208E+01	4.0125050	.91111805E-01	9.98197571	-23.38978902	.22913E+00	.43392E+02	.89084E-03	9.18	-4.00
-.18439668E+01	.28677876E+01	4.3998031	.11371193E+00	11.46965836	-.22945551	.19204E+00	.36530E+02	.74995E-03	11.47	-.05
-.93729636E+00	.34452250E+01	4.8271494	.14249609E+00	12.99360365	17.26918725	.15945E+00	.30500E+02	.62616E-03	12.43	3.92
.73534036E-01	.37766042E+01	5.2317488	.17351619E+00	14.30242924	30.20142917	.13469E+00	.25920E+02	.53213E-03	12.43	7.31
.14855214E+01	.38634369E+01	5.6408601	.20858389E+00	16.13727865	44.30570715	.11426E+00	.22140E+02	.45453E-03	11.70	11.43
.28362590E+01	.34402283E+01	5.9546958	.23794598E+00	17.80305172	57.04705689	.10107E+00	.19699E+02	.40442E-03	9.91	15.08
.39578664E+01	.25830551E+01	6.1511881	.25738007E+00	19.03489716	68.30124844	.93721E-01	.18340E+02	.37651E-03	7.27	17.77
.47057545E+01	.13865234E+01	6.2708383	.26959763E+00	19.80133052	79.20398652	.89550E-01	.17568E+02	.36066E-03	3.88	19.48
.49686030E+01	-.24067052E-11	6.3103563	.27369523E+00	20.06426459	90.00000000	.88219E-01	.17322E+02	.35561E-03	0.00	20.06

TO USE AS IVS, PUNCH THE 3 CARDS, PLACE BEFORE BODY PT CARDS

```
   47    5    0    0    0    0    0    0   11
 0.              .10000000E+01 0.              .10000000E+01
  .19658036E+00 -.10000000E+00  .64024979E+01
```

DATA POINTS WILL BE RE-SPACED FOR SURFACE 48

FIGURE 18. DATA SURFACE WHERE CFL CONDITION IS VIOLATED

DATA SURFACE NO 30

***TO USE THIS AS AN INITIAL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS

SURFACE NO 30. STATION Y = .31060794E+01

BODY POINTS

------Z-------	------X------	MACH NO.	----PT/PT0----	THETA, DEG	--PSI,DEG--	-----CP-----	----P/P0----	---P/PT0---	OUTWASH	UPWASH
-.10000000E+01	-.18367530E-11	2.5286160	.36828400E-02	0.00000000	0.00000000	.14978E+00	.10663E+02	.32965E-03	-.00	.00
-.10000000E+01	.22415743E+00	2.2272640	.36828400E-02	1.15947403	.00000000	.14997E+00	.10675E+02	.33004E-03	1.16	.00
-.10000000E+01	.45417705E+00	2.2332919	.36828400E-02	2.12331815	.00000000	.14842E+00	.10575E+02	.32694E-03	2.12	.00
-.10000000E+01	.68735226E+00	2.2361681	.36828400E-02	2.94828163	.00000000	.14768E+00	.10527E+02	.32547E-03	2.95	.00
-.98685729E+00	.93851740E+00	2.1456040	.36828400E-02	6.56735715	-19.99078936	.17253E+00	.12130E+02	.37502E-03	6.17	-2.25
-.91755860E+00	.11896681E+01	2.1445243	.36828400E-02	11.65209318	-19.57399363	.20468E+00	.14205E+02	.43915E-03	11.00	-3.95
-.78675445E+00	.14234312E+01	1.9764114	.36828400E-02	16.47186625	-10.52622654	.22931E+00	.15793E+02	.48826E-03	16.21	-3.09
-.59762294E+00	.16197838E+01	1.9528439	.36828400E-02	20.42728125	1.61391623	.23606E+00	.16228E+02	.50172E-03	20.42	.60
-.36151837E+00	.17587523E+01	1.9975021	.36828400E-02	22.99855999	15.14008193	.22141E+00	.15284E+02	.47252E-03	22.28	6.33
-.99073259E-01	.18254424E+01	2.0406825	.36828400E-02	23.94101674	29.34499715	.18938E+00	.13217E+02	.40863E-03	21.16	12.28
.16536994E+00	.18160762E+01	2.0278891	.36828400E-02	23.31957768	43.89456893	.14941E+00	.10665E+02	.32971E-03	17.26	16.64
.41006148E+00	.17371419E+01	2.4030037	.36828400E-02	21.38062615	58.57412253	.11021E+00	.81098E+01	.25073E-03	11.54	18.47
.61542054E+00	.16053340E+01	2.5585143	.36828400E-02	18.26440236	71.70255117	.83181E-01	.63662E+01	.19682E-03	5.92	17.40
.89803423E+00	.12347093E+01	2.8995776	.36828400E-02	11.45736964	105.61107541	.42981E-01	.37728E+01	.11664E-03	-3.32	10.99
.99972432E+00	.79153662E+00	[illegible]	.36828400E-02	7.57335377	175.27283153	.19742E-01	.22736E+01	.70291E-04	-7.55	.63

ILLNATURED PSI AND DELTA.CASE NO 2. ROTATE COORDINATES AND RECOMPUTE SURFACE 30

FIGURE 19. DATA SURFACE WITH UNDEFINED FLOW ANGLES

```
                                        DATA SURFACE NO  5

***TO USE THIS AS AN INITIAL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS

SURFACE NO  5, STATION Y =   8.07151645E-01

                                               BODY POINTS
------Z-------  ------X------  MACH NO.  ----PT/PTO---- THETA, DEG  --PSI,DEG--   -----CP-----  ----P/PO---- ---P/PTO---OUTWASHUPWASH
-9.81228551E-01-1.84653521E-12 1.6642122 3.68284000E-03 11.11905779 -90.00000000  3.79277E-01  2.54679E+01  7.87376E-04  0.00-11.12
-9.45324351E-01 2.66429836E-01 1.6659499 3.68284000E-03 12.51305080 -54.54695476  3.78250E-01  2.54016E+01  7.85327E-04  7.34-10.25
-8.40328774E-01 5.13212298E-01 1.6769635 3.68284000E-03 15.77939305 -27.66582158  3.71793E-01  2.49851E+01  7.72449E-04 14.05 -7.47
-6.74253379E-01 7.22161737E-01 1.7063047 3.68284000E-03 19.46817102  -7.33251556  3.55049E-01  2.39050E+01  7.39055E-04 19.32 -2.58
-4.59802261E-01 8.78117382E-01 1.7608553 3.68284000E-03 22.60173159  10.01947099  3.25642E-01  2.20078E+01  6.80402E-04 22.29  4.14
-2.13582269E-01 9.70168483E-01 1.8504238 3.68284000E-03 24.56732191  26.13547407  2.81955E-01  1.91895E+01  5.93269E-04 22.31 11.39
 4.53504553E-02 9.92932165E-01 1.9736798 3.68284000E-03 25.00873945  41.47269503  2.30347E-01  1.58602E+01  4.90339E-04 19.27 17.17
 2.97373639E-01 9.47239109E-01 2.1208852 3.68284000E-03 23.84907867  55.84664083  1.79978E-01  1.26081E+01  3.89798E-04 13.94 20.09
 5.26269663E-01 8.38919959E-01 2.2722019 3.68284000E-03 21.46987910  69.46770017  1.38735E-01  9.95009E+00  3.07621E-04  7.85 20.22
 7.18143931E-01 6.77320142E-01 2.4222044 3.68284000E-03 18.37633264  81.92463865  1.06487E-01  7.87037E+00  2.43323E-04  2.67 18.21
 8.62124735E-01 4.74641263E-01 2.5658646 3.68284000E-03 15.04289334  91.10298432  8.20619E-02  6.29398E+00  1.94587E-04  -.30 15.04
 9.51114521E-01 2.44355848E-01 2.6545407 3.68284000E-03 12.24551017  93.84459345  6.95589E-02  5.48738E+00  1.69650E-04  -.83 12.22
 9.81228551E-01-2.12794317E-12 2.6824162 3.68284000E-03 11.11905779  90.00000000  6.59822E-02  5.25664E+00  1.62516E-04  0.00 11.12

                                               FIELD POINTS
------Z-------  ------X------  MACH NO.  ----PT/PTO---- THETA, DEG  --PSI,DEG--   -----CP-----  ----P/PO---- ---P/PTO---OUTWASHUPWASH
-1.07775920E+00-2.04731492E-12 1.7356564 5.04201119E-03 14.08094923 -90.00000000  4.69766E-01  3.13055E+01  9.67855E-04  0.00-14.08
-1.03872179E+00 2.96479622E-01 1.7485827 5.07835754E-03 15.12438295 -57.23524886  4.63769E-01  3.09187E+01  9.55893E-04  8.32-12.80
-9.24852294E-01 5.72200721E-01 1.7858764 5.16255436E-03 17.66249977 -30.00262585  4.44816E-01  2.96959E+01  9.18091E-04 15.42 -9.05
-7.44859700E-01 8.08311276E-01 1.8448233 5.24814465E-03 20.60240689  -8.56076772  4.12046E-01  2.75819E+01  8.52733E-04 20.39 -3.20
-5.11932148E-01 9.89399158E-01 1.9272160 5.36487336E-03 23.17282960   9.00343455  3.69388E-01  2.48300E+01  7.67653E-04 22.92  3.83
-2.42659277E-01 1.10479261E+00 2.0270964 5.46344348E-03 25.04236698  24.12141110  3.20135E-01  2.16525E+01  6.69418E-04 23.09 10.81
 4.47665102E-02 1.14878990E+00 2.1359048 5.53707371E-03 26.17490998  37.32110469  2.71517E-01  1.85161E+01  5.72450E-04 21.35 16.59
 3.54079954E-01 1.11762657E+00 2.2527919 5.62523452E-03 26.82663603  49.44057901  2.27349E-01  1.56667E+01  4.84359E-04 18.20 21.02
 6.42969408E-01 1.00707155E+00 2.3583103 5.68871105E-03 27.07912372  59.74141471  1.92719E-01  1.34327E+01  4.15289E-04 14.45 23.83
 8.91724014E-01 8.25354393E-01 2.4471812 5.73190943E-03 27.04237170  68.63888321  1.67121E-01  1.17813E+01  3.64236E-04 10.53 25.43
 1.08321043E+00 5.85221382E-01 2.5147090 5.77306786E-03 26.85368789  76.43722245  1.50078E-01  1.06818E+01  3.30243E-04  6.77 26.21
 1.20408941E+00 3.03494585E-01 2.5549273 5.79715621E-03 26.72997615  83.41741731  1.40701E-01  1.00769E+01  3.11542E-04  3.30 26.58
 1.24543403E+00-2.71251019E-12 2.5681889 5.80436809E-03 26.68878839  90.00000000  1.37712E-01  9.88407E+00  3.05580E-04  0.00 26.69

                                               FIELD POINTS
------Z-------  ------X------  MACH NO.  ----PT/PTO---- THETA, DEG  --PSI,DEG--   -----CP-----  ----P/PO---- ---P/PTO---OUTWASHUPWASH
-1.17201953E+00-2.24809409E-12 1.9247629 8.21024810E-03 15.74755014 -90.00000000  5.75743E-01  3.81436E+01  1.17926E-03  0.00-15.75
-1.13000869E+00 3.25837204E-01 1.9453135 8.33494379E-03 16.65910707 -58.52985006  5.65936E-01  3.75097E+01  1.15966E-03  8.88-14.32
-1.00742337E+00 6.29797846E-01 2.0023222 8.67943557E-03 18.94546481 -31.75801510  5.38665E-01  3.57503E+01  1.10527E-03 16.27-10.24
-8.13509741E-01 8.92286972E-01 2.0904024 9.22261093E-03 21.70605331 -10.35407765  4.97793E-01  3.31136E+01  1.02375E-03 21.38 -4.09
-5.62301697E-01 1.09787115E+00 2.1976071 9.89025786E-03 24.35180453   6.84129057  4.49906E-01  3.00301E+01  9.28424E-04 24.20  3.09
-2.70364390E-01 1.23573360E+00 2.3143923 1.06585717E-02 26.62142609  21.08871630  4.02360E-01  2.69571E+01  8.33415E-04 25.06 10.22
 4.45537223E-02 1.29938606E+00 2.4338490 1.15015489E-02 28.50563041  33.34984867  3.58643E-01  2.41368E+01  7.46222E-04 24.40 16.62
 4.08718531E-01 1.28064433E+00 2.5616372 1.24820884E-02 30.08444015  45.12792737  3.17342E-01  2.14724E+01  6.63848E-04 22.23 22.32
 7.53622080E-01 1.16575685E+00 2.6724092 1.33265093E-02 31.31154551  55.56728029  2.83939E-01  1.93168E+01  5.97206E-04 18.98 26.64
 1.05427399E+00 9.63156776E-01 2.7642736 1.41517593E-02 32.18607932  64.95987270  2.58524E-01  1.76779E+01  5.46536E-04 14.92 29.69
 1.28819841E+00 6.87075781E-01 2.8356295 1.46943115E-02 32.75701932  73.65148231  2.41628E-01  1.65879E+01  5.12838E-04 10.27 31.69
 1.43695126E+00 3.57651372E-01 2.8744054 1.50288385E-02 33.08659402  81.92146836  2.32433E-01  1.59947E+01  4.94499E-04  5.23 32.83
 1.48801870E+00-3.10994824E-12 2.8878993 1.51626742E-02 33.19655424  90.00000000  2.29575E-01  1.58103E+01  4.88798E-04  0.00 33.20

                                               SHOCK POINTS
------Z-------  ------X------  MACH NO.  ----PT/PTO---- THETA, DEG  --PSI,DEG--   -----CP-----  ----P/PO---- ---P/PTO---OUTWASHUPWASH
-1.29816536E+00-2.44889140E-12 2.4016723 2.18998062E-02 17.58265787 -90.00000000  7.33581E-01  4.83248E+01  1.49403E-03  0.00-17.58
-1.25290101E+00 3.60228366E-01 2.4290929 2.25462477E-02 18.08226563 -65.77963588  7.23387E-01  4.76671E+01  1.47370E-03  7.63-16.58
-1.12044986E+00 6.97857585E-01 2.5061201 2.44596203E-02 19.48770903 -43.30385732  6.95463E-01  4.58657E+01  1.41800E-03 14.44-13.64
-9.10158024E-01 9.92485515E-01 2.6251149 2.77145894E-02 21.51789859 -23.47653244  6.54349E-01  4.32133E+01  1.33600E-03 19.88 -8.93
-6.36201933E-01 1.22775314E+00 2.7693005 3.21868641E-02 23.91007501  -6.35850576  6.07732E-01  4.02053E+01  1.24300E-03 23.78 -2.81
-3.15097752E-01 1.39144865E+00 2.9532312 3.88251256E-02 26.26071826   8.82776383  5.53044E-01  3.66780E+01  1.13395E-03 25.99  4.33
 5.00904083E-02 1.47936713E+00 3.1090884 4.53607805E-02 31.79524331  32.05639748  5.10650E-01  3.39431E+01  1.04940E-03 27.72 18.21
 4.61531596E-01 1.47166220E+00 3.2349866 5.13115494E-02 31.59276518  34.96642906  4.78858E-01  3.18921E+01  9.85988E-04 26.75 19.42
 8.70425496E-01 1.35071047E+00 3.4183113 6.11506906E-02 33.54841365  47.20611176  4.36187E-01  2.91393E+01  9.00882E-04 24.25 25.95
 1.23070682E+00 1.12335796E+00 3.5473210 6.49780677E-02 35.27317695  58.55689115  4.08541E-01  2.73558E+01  8.45743E-04 20.25 31.11
 1.51322570E+00 8.05204630E-01 3.6451097 7.54440953E-02 36.50742274  69.33182359  3.88802E-01  2.60824E+01  8.06373E-04 14.64 34.70
 1.69398685E+00 4.20409471E-01 3.7042376 7.95871117E-02 37.25843303  79.74367297  3.77349E-01  2.53435E+01  7.83530E-04  7.71 36.81
 1.75624758E+00-3.49444749E-12 3.7239540 8.10087902E-02 37.51140104  90.00000000  3.73608E-01  2.51022E+01  7.76068E-04  0.00 37.51

TO USE AS IVS, PUNCH THESE CARDS, PLACE BEFORE BODY PT CARDS
   5    4    0    0    0    0    0    0   13
 0.              1.00000000E+00 0.              1.00000000E+00
 4.00649605E-02 5.00000000E-02 8.07151645E-01

SURFACE NO  6, STATION Y =  8.49219854E-01  LOWER CENTERLINE CP =  3.38154772E-01  UPPER CENTERLINE CP =  5.83298728E-02

SURFACE NO  7, STATION Y =  8.93391473E-01  LOWER CENTERLINE CP =  3.00044858E-01  UPPER CENTERLINE CP =  5.12612274E-02

SURFACE NO  8, STATION Y =  9.39771673E-01  LOWER CENTERLINE CP =  2.64572959E-01  UPPER CENTERLINE CP =  4.47370136E-02

SURFACE NO  9, STATION Y =  9.88470883E-01  LOWER CENTERLINE CP =  2.31530062E-01  UPPER CENTERLINE CP =  3.87227025E-02
```

FIGURE 20. SPECIAL OPTION PRINT

SECTION V

OPERATIONAL ASPECTS

1. CORE AND TIME REQUIREMENTS

The program requires a core size of $60,000_8$ to compile, $1000,000_8$ to load, and $61,300_8$ to execute, all on the CDC 6600 computer.

The program requires approximately 30 seconds of Central Processor Unit (CPU) time to compile on the FTN compiler, using OPT = 1. A typical run will require approximately 10 minutes of CPU time for the fuselage of Phase I and one hour for the wing-body of Phase II. There is practically no difference in the execution times generated by the OPT = 1 and OPT = 2 compiler options. Therefore, OPT = 1 is recommended because of its somewhat faster compilation.

2. START OPTIONS

a. *Normal Start*

Provisions have been made to run both the Initial Value Surface Program and the Three-Dimensional Method of Characteristics Program in one pass. This is done by combining the control cards; putting the Initial Value Surface Program and data before the Three-Dimensional Method of Characteristics Program and data, and setting ISTART = 0 and IFT = 1.

During execution the 3DMoC program will automatically read the initial value surface from disc file TAPE4 where it was placed by the IVS program. It is also possible to run the IVS program and the 3DMoC program separately, using a physical tape to transfer the IVS information. In this case the IVS data on TAPE4 are recorded on a physical tape through a COPYCF card in the Initial Value Surface Program at the end of the computation. In the Three-Dimensional Method of Characteristics Program, set IFT = 1 and ISTART = 0 and transfer the IVS data from the physical tape to disc file TAPE4 through a COPYCF card before computation begins. The physical tape (TAPE1) on which all data surfaces will be written, can double up for the job of transferring the IVS data.

b. Restart

Normal output can either be written on a physical tape (TAPE1) or be printed out. When a restart is made from the Nth data surface on the tape, set ISTART = N. Be sure that ISTART $\neq 0$, otherwise the data stored on the tape will be ruined.

When the normal output is printed out, a restart can be made by one of the following methods. When the output control KP4 is set to zero, every 10th data surface will be written on TAPE1, which can then be used for a restart by setting ISTART = m(10). A restart can be made from any completed data surface by key-punching the data cards from the listing and making a restart from cards with ISTART set equal to any negative integer. In key-punching the data, ignore any line with literal information; simply key-punch the first six columns of data and stack the last four cards first, as described in Section III. During restart, the number of data rings per data surface can be increased. Set INCRES = K for restarting from cards or set IFT = -K for restarting from tape. The number of data rings N will be increased to N+K. It is recommended to limit K to 2 or less.

3. DEFAULT PRINT

A printout option is built into the program to print intermediate results at a particular point, along a particular data ring or on a particular data surface at the lower level KPRINT = 1 or at the higher level KPRINT =2. This may be done for check-out purposes when other means have failed to reveal a source of computational trouble. Level 2 gives extensive default print, while level 1 gives a moderate amount of default print. As a rule, up to level 2 is used at a data point, while level 1 is used along a data ring. Otherwise, the amount of printout obtained could be extensive.

In case a failure occurs during a restart from a physical tape, it is useful to keep track of the data that are stored on the tape. For this purpose, KPRINT(3) can be set equal to 3 to print out the first 40 columns of the stored data as they are read in during the restart. Other intermediate results will be printed out at the higher level 2 at point NP, along ring NL, on data surface NS when KPRINT(3) is set to 3.

4. ERROR MESSAGES

If the program runs all the way, it is a fairly good assumption that the results are good. Usually, if something goes wrong, the computation will stop sooner or later with an error message. Default print can be obtained, if necessary, to assist in locating the source of the error. In the case that the normal output is on TAPE1, and the computation stops at a data surface which is not printed out, it will be necessary to restart from the previous data surface and set KP4 = 1 to have the partial data surface printed out. This will locate the troublesome data point and detailed default print may then be obtained for that point.

Error messages, unusual stops and other program messages are summarized below below. They are grouped according to the program or subroutine in which they are generated.

Program FUSLAG

ILLNATURED PSI AND DELTA CASE NO __, ROTATE COORDINATES AND RECOMPUTE SURFACE

Whenever the local velocity is aligned with the Y-axis, the angles ψ and δ are undefined. When this happens, the coordinate system is rotated around the Z-axis through an angle of 0.1 radian. The above message is printed out (for the Ith such occurrence) and the Nth data surface is recomputed.

DATA POINTS WILL BE RE-SPACED FOR SURFACE __

This message will appear when (1) the distribution of data points along a body ring becomes noticeably uneven, (2) a data ring is automatically added by the program, (3) a data point computation violates the CFL stability condition

WRONG IVS NO ___, NO ___ SPECIFIED OR FAIL END FILE

When a restart is made from TAPE1, this message indicates either the program is unable to locate and read from the specified data surface or the arrangement of data on TAPE1 is unusual so that an END OF FILE was encountered. If this happens, the contents of TAPE1 should be printed out. Alternatively, when KPRINT(3) is set to 3, the first 40 columns of each line information on TAPE1 will be printed out as the program FUSLAG is reading TAPE1.

Subroutine BULK

BULK POINT HAS FAILED TO CONVERGE AFTER 25 ITERATIONS

Computation is stopped.
This happens very rarely. A decrease in the magnitude of the convergence criteria may remedy the problem.

Subroutine BASEPT

BASE POINT ___ VIOLATES CFL

This advisory message appears whenever the CFL stability condition is violated at basepoint I. The number following the message is the factor L (see the discussion on basepoint location in Volume I).

INTERSECTION FAILURE, BASEPOINT NO ___

Computation is stopped.
This message indicates that the Mach conoid fails to properly intersect the line joining basepoint I to the center data point. The first three numbers following the message are coefficients A, B, C, of the quadratic equation (see discussion of basepoint location in Volume I) and the last number is $\sqrt{B^2 - AC}$. When this happens at the juncture point of two body sections check the continuity of the body description. If the body description is new, check FIGURE, paying attention to the formula and coding of the body normal. In high gradient areas, this message usually means more data points are required.

MACH CONE CONVERGENCE FAILURE IN BASEPOINT NO ___

Computation is stopped.
This seldom happens. A relaxing of the convergence criterion on Mach angle (increase in the value of EPSLN) helps.

BASE PT ___ EXTRAPOLATES, STEP DECREASED

The number following this message is the factor L (see discussion on basepoint location in Volume I). If this message appears in a series prior to an abnormal termination, an increase in the number of data rings (and in some cases, in the number of data points) may help.

Subroutine COMPAT

CONVERGENCE FAILURE IN COMPAT

Computation is stopped.
This has never happened. If it should occur, try to increase the number of iterations by modifying the source deck. If it should persist, obtain default print for analysis of the problem.

Subroutine GMTRY

CONVERGENCE FAILURE IN GMTRY

Computation is stopped.
Check the body description and subroutine FIGURE.

Subroutine FIGURE

$$$$$ YOU HAVE GONE ALL THE WAY--FEELING NICE, I BET $$$$$

This message indicates that the goal has been accomplished

SECTION VI

LOGICAL STRUCTURE

1. INTERDEPENDENCE OF SUBROUTINES

The Calling-Called matrix for the program is shown in Figure 21.

2. LISTINGS

Two complete listings of the Three-Dimensional Method of Characteristics Program are given below, following Figure 21. The first listing is for the computer program developed in Phase I for the calculation of flow fields over a fuselage and canopy configuration. The second listing is for the computer program developed in Phase II for the calculation of flow fields about a fuselage-canopy-wing configuration.

Calling \ Called	BULK	PICK	BASEPT	COEFS	COMPAT	GMTRY	FIGURE	TRANS	ROTATE	SOL	HARNES	MIRROR	NORM	DATOUT	TIDYUP	REFORM	KURFIT	KURVE	ROTXY	CONICF	CUBIC	ELIPSE	WINGF	PCRCL	PELIPS	CONPIC	XSUBG	AERO	SETCP	SECOND
FUSLAG	●								●		●	●		●	●	●												●	●	
BULK		●	●	●	●	●		●		●			●																	
PICK													●																	
BASEPT													●																	
COEFS										●			●																	
GMTRY							●		●				●																	
TRANS									●																					
HARNES													●																	
TIDYUP									●								●	●												●
REFORM																	●	●												
FIGURE																			●	●	●	●	●	●	●	●	●			
CONPIC																										●				

FIGURE 21. CALLING-CALLED MATRIX FOR 3DMoC PROGRAM

```
*DECK A
      PROGRAM FUSLAG(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PUNCH,TAPE7=    A   1
     1          PUNCH,TAPE2,TAPE1,TAPE4)                                    A   2
C **** FUSELAGE + CANOPY (PHASE-I CONTRACT)
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST,KPRINT(3)
      COMMON /GEOM/ YMAX,YSTOP,INDEX                                       H0030
      COMMON /MESH/ K1,K2                                                  A   5
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3     A   6
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)                  A   7
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE,ISURF,Y0,ISTART          A   8
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,   A   9
     1PS,HUB(8)                                                            A  10
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),     A  11
     1PPMBS(4),THBS(4),PSBS(4)                                             A  12
      COMMON /STOR/ H(4,8),A(50,8,4)                                       A  13
      COMMON /CP/B(50,4,2),DRAG,ALIFT,TORQUE,YMT,ZMT,WINGA
      DIMENSION YSET(10),NPSET(30),LIT(18),DUM(10)                         A  14
C     IN A(I,J,K) J=1,8 CORRESPOND TO X,Y,Z,U,V,W,P,PT, K=1,4 CORRESPOND  A  15
C     TO A(ACQUIRER), L(LOWER CLASS), M(MIDDLE CLASS), U(UPPER CLASS)    A  16
      CALL FTNBIN (1,0,DUM)                                                A  17
      IOU5=5                                                               A  18
      IOU6=6                                                               A  19
      IOU7=7                                                               A  20
      IOU3=1                                                               A  21
      IOU=IOU5                                                             A  22
      LTAPE=4                                                              A  23
      NTAPE=2                                                              A  24
      REWIND LTAPE                                                         A  25
      REWIND NTAPE                                                         A  26
      LN=LTAPE+NTAPE                                                       A  27
      READ (IOU5,52) (LIT(I),I=1,18)                                       A  28
      READ (IOU5,55) DP,DTHETA,DPSI,EPSLN,DYC,YSTOP,STEPM
      READ (IOU5,56) NSURF,LINE,NPC,NR,IFT,ISTART,KP4,NS,NL,NP,KPRINT     A  30
      IOU1=IOU3                                                            A  31
      IF (KP4.LT.0) IOU1=IOU6                                              A  32
      KAB=IABS(KP4)                                                        A  33
      IF (KP4.EQ.0) KAB=10                                                 A  34
      DO 1 I=1,NR                                                          A  35
    1 READ (IOU5,53) YSET(I),NPSET(3*I-2),NPSET(3*I-1),NPSET(3*I)         A  36
      READ (IOU5,55) FMSTR,GF(1),ALFA,Y0                                   A  37
      READ (IOU5,55) YA,ZA,SCALE,WINGA                                     A  38
      WRITE (IOU6,54) (LIT(I),I=1,18)                                      A  39
      WRITE(IOU6,55) DP,DTHETA,DPSI,EPSLN,DYC,YSTOP,STEPM
```

```
      WRITE(IOU6,56) NSURF,LINE,NPC,NR,IFT,ISTART,KP4,NS,NL,NP,KPRINT
      DO 105 I=1,NR
105   WRITE(IOU6,53) YSET(I),NPSET(3*I-2),NPSET(3*I-1),NPSET(3*I)
      WRITE(IOU6,55) FMSTR,GF(1),ALFA,Y0
      WRITE(IOU6,55) YA,ZA,SCALE,WINGA
      WINGA=WINGA*144.0
      STEP=0.
      MESS=1
      INCRES=0
      GF(2)=.5*(GF(1)-1.)
      GF(3)=.5*(GF(1)+1.)
      GF(4)=GF(2)/GF(3)
      GF(5)=1./(GF(1)-1.)
      GF(6)=GF(1)*GF(5)
      GF(7)=1./GF(6)
      GFM2=GF(2)*FMSTR*FMSTR
      QINF=SQRT(GFM2/(1.+GFM2))
      POP=(1.+GFM2)**GF(6)
      RAD=57.295779513082
      COSA=COS(ALFA/RAD)
      SINA=SIN(ALFA/RAD)
      ISURF=1
      ILL=0
      KURE=0
      K1=0
      K2=0
      SINE=0.
      COSINE=1.
      S2=0.
      C2=1.
      SN=SIN(.1)
      CS=COS(.1)
      YMAX=-1.E7
C ISTART.LT.0,RESTART FROM CARDS,ISTART=I.GT.0,RESTART FROM ITH SURFACE
      IF (ISTART.NE.0.OR.IFT.NE.1) GO TO 2
      IOU=LTAPE
      REWIND IOU3
      READ (IOU,55) FMSTR,GF(1),ALFA,YI,DRAG $ Y0=(YI-1.0)*SCALE+YA
      DRAG = DRAG*3.1415927*SCALE**2/WINGA
2     IF (ISTART.EQ.0) REWIND IOU3
      IF (ISTART.NE.0.OR.KP4.LE.0) GO TO 3
      WRITE (IOU3,68) (LIT(I),I=1,18),FMSTR,GF(1),ALFA
      WRITE (IOU3,62) ISURF,LINE,ILL,KURE,MESS,INCRES,K1,K2,NPC
```

```
      WRITE (IOU3,66) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE       A  77
    3 IF (ISTART) 8,9,4                                                     A  78
    4 DO 6 I=1,ISTART                                                       A  79
      READ (IOU3,52) DUM                                                    A  80
      IF (KPRINT(3).EQ.3) WRITE (IOU6,52) DUM                               A  81
      READ (IOU3,65) ISURF,LINE,ILL,KURE,MESS,INCRES,K1,K2,NPC              A  82
      IF (KPRINT(3).NE.0) WRITE (IOU6,56) ISURF,LINE,K1,K2,NPC              A  83
      READ (IOU3,66) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE        A  84
      READ (IOU3,52) DUM                                                    A  85
      IF (IFT.LT.0) INCRES=-IFT                                            A 855
      IF(IFT.GT.1) INCRES=1-IFT
      IF (ISURF.EQ.ISTART) GO TO 9                                          A  86
      KOUNT=LINE*(NPC+2)+100                                                A  87
      DO 5 J=1,KOUNT                                                        A  88
      READ (IOU3,52) DUM                                                    A  89
      IF (KPRINT(3).EQ.3) WRITE (IOU6,52) DUM                               A  90
      IF (EOF(IOU3)) 6,5                                                    A  91
    5 CONTINUE                                                              A  92
    6 CONTINUE                                                              A  93
    7 WRITE (IOU6,64) ISURF,ISTART                                          A  94
      CALL EXIT                                                             A  95
    8 READ (IOU5,56) ISURF,LINE,ILL,KURE,MESS,INCRES,K1,K2,NPC              A  96
      READ (IOU5,57) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE        A  97
    9 N1=ISURF+1                                                            A  98
      N2=NPC+1                                                              A  99
      N3=NPC+3                                                              A 100
      WRITE(IOU6,58) FMSTR,GF(1),ALFA,DRAG,ALIFT,TORQUE
      DO 13 I=1,LINE                                                        A 101
      IF (ISTART.GT.0) READ (IOU3,52) DUM                                   A 102
      IND=2                                                                 A 103
      IF (I.EQ.1) IND=1                                                     A 104
      IF (I.EQ.LINE) IND=3                                                  A 105
      IF (ISTART.GT.0) READ (IOU3,52) DUM                                   A 106
      DO 12 J=2,N2                                                          A 107
      IF (ISTART.LE.0) READ (IOU,59) A(J,3),A(J,1),A(J,7),A(J,8),THETA,     A 108
     1PSI                                                                   A 109
      IF (ISTART.GT.0) READ (IOU3,67) A(J,3),A(J,1),A(J,7),A(J,8),THETA,    A 110
     1PSI                                                                   A 111
      THETA=THETA/RAD                                                       A 112
      PSI=PSI/RAD                                                           A 113
      A(J,7)=A(J,8)/(1.+GF(2)*A(J,7)**2)**GF(6)                             A 114
      IF (ISTART.NE.0) GO TO 10                                             A 115
      A(J,1)=A(J,1)*SCALE                                                   A 116
```

```
      A(J,3)=A(J,3)*SCALE+ZA                                            A 118
   10 A(J,2)=Y0                                                         A 119
      Q=SQRT(1.-(A(J,7)/A(J,8))**GF(7))                                 A 120
      QS=Q*SIN(THETA)                                                   A 121
      A(J,4)=QS*COS(PSI)                                                A 122
      A(J,5)=Q*COS(THETA)                                               A 123
      A(J,6)=QS*SIN(PSI)                                                A 124
      DO 11 K=1,4,3                                                     A 125
   11 CALL ROTATE (A(J,K),A(J,K+1),SINE,COSINE)                         A 126
      CALL DATOUT (A,J,IND,KAB,0)                                       A 127
   12 CALL DATOUT (A,J,IND,KAB,1)                                       A 128
   13 WRITE (NTAPE) ((A(J,L),L=1,8),J=2,N2)                             A 129
      IF (ISTART.EQ.0.AND.KP4.GT.0) END FILE IOU3                       A 130
      IF (ISTART.LE.0) GO TO 14                                         A 131
      READ (IOU3,52) DUM                                                A 132
      IF (EOF(IOU3)) 14,7                                               A 133
   14 DO 51 I=N1,NSURF                                                  A 134
      ITEST=0                                                           A 135
      KTEST=0                                                           A 136
      ISURF=I                                                           A 137
      NLINE=LINE+INCRES                                                 A 138
   15 DYC=(DYC+DYC*STEP)*FLOAT(LINE)/FLOAT(NLINE)                       A 139
      IF (STEP.LT.0.D0) MESS=1                                          A 140
      STEP=.05                                                          A 141
      IPRINT=0                                                          A142
      DO 16 K=1,NR                                                      A 143
      NPCC=NPSET(3*(K-1)+1)                                             A 144
      L1=NPSET(3*(K-1)+2)                                               A 145
      L2=NPSET(3*(K-1)+3)                                               A 146
      IF (Y0+DYC.LT.YSET(K).OR.NPSET(3*K+1).EQ.0) GO TO 17              A147
   16 CONTINUE                                                          A 148
   17 IF (NPC.EQ.NPCC.AND.(L1.EQ.K1.AND.L2.EQ.K2)) GO TO 18             A 149
      DYC=DYC*FLOAT(NPC-1-(K2-K1)/2)/FLOAT(NPCC-1-(L2-L1)/2)            A 150
      K1=L1                                                             A 151
      K2=L2                                                             A 152
      MESS=1                                                            A 153
   18 REWIND LTAPE                                                      A 154
      REWIND NTAPE                                                      A 155
      IF((Y0+DYC.LE.YSTOP).OR.(ABS(YSTOP-Y0).LT.0.0002)) GO TO 181
      DYC=YSTOP-Y0
      KP4=-10000
  181 IF (ITEST.EQ.KTEST) GO TO 19
      IF (ITEST.GT.10) CALL EXIT                                        A 157
```

```
      KTEST=ITEST
      GO TO 26
19    LTAPE=LN-LTAPE
      NTAPE=LN-NTAPE
      IF (MESS.EQ.0.AND.INCRES.EQ.0) GO TO 21
      DO 20 K=1,LINE
      READ (LTAPE) ((A(L,M),M=1,8),L=1,NPC)
      IF (K.EQ.1) AM=.5*(A(1,3)+A(NPC,3))
      CALL TIDYUP (SINE,COSINE,A,NPC,NPCC,AM)
20    WRITE (NTAPE) ((A(L,M,2),M=1,8),L=1,NPCC)
      WRITE (IOU6,63) I
      NPC=NPCC
      N2=NPC+1
      N3=NPC+3
      CALL REFORM (A,LTAPE,NTAPE,NPC,LINE,NLINE)
      INCRES=0
      MESS=0
      GO TO 18
21    IF (KURE.NE.ILL) 23,26
22    REWIND LTAPE
      REWIND NTAPE
23    WRITE (IOU6,61) ILL,I
      IF (ILL.GT.16) CALL EXIT
      KURE=ILL
      CALL ROTATE (SINE,COSINE,SN,CS)
      S2=2.*SINE*COSINE
      C2=COSINE*COSINE-SINE*SINE
      DO 25 J=1,LINE
      READ (LTAPE) ((A(L,K),K=1,8),L=2,N2)
      DO 24 K=2,N2
      DO 24 L=1,4,3
24    CALL ROTATE (A(K,L),A(K,L+1),SN,CS)
25    WRITE (NTAPE) ((A(L,K),K=1,8),L=2,N2)
      REWIND LTAPE
      REWIND NTAPE
      LTAPE=LN-LTAPE
      NTAPE=LN-NTAPE
26    IF (KP4.LE.1.OR.I/KP4*KP4.EQ.I) WRITE (IOU6,60) I
      CALL SETCP(LTAPE,1,A,B,SINE,COSINE,POP,GF,FMSTR,N2)
      IF(STEPM.NE.0.) DYC=AMIN1(DYC,STEPM)
      DO 45 J=1,LINE
      IND=2
      IF (J.EQ.1) IND=1
```

```
      IF (J.EQ.LINE) IND=3
      IF (IND-2) 27,31,34
27    DO 28 M=3,4
28    READ (LTAPE) ((A(K,L,M),L=1,8),K=2,N2)
      IF (DYC.GT.1.E-6) GO TO 29
      DYC=AMIN1(ABS(A(2,3,3)-A(2,3,4)),ABS(A(3,1,3)-A(2,1,3)))
      XM2=(1.*(A(2,8,3)/A(2,7,3))**GF(7)-1.)/GF(2)
      DYC=DYC*.6*SQRT(XM2-1.)
29    CALL MIRROR (A(1,1,3),S2,C2,N2)
      DO 30 K=2,NPC
      IF (.6*((A(K,1,4)-A(K,1,3))**2+(A(K,2,4)-A(K,2,3))**2+(A(K,3,4)-A(
     1K,3,3))**2).LT.(A(K,1,3)-A(K+1,1,3))**2+(A(K,2,3)-A(K+1,2,3))**2+(
     2A(K,3,3)-A(K+1,3,3))**2) GO TO 33
30    CONTINUE
      INCRES=1
      GO TO 33
31    DO 32 K=1,N3
      DO 32 L=1,8
      A(K,L,2)=A(K,L,3)
32    A(K,L,3)=A(K,L,4)
      READ (LTAPE) ((A(L,K,4),K=1,8),L=2,N2)
33    CALL MIRROR (A(1,1,4),S2,C2,N2)
      GO TO 36
34    DO 35 K=3,4
      DO 35 L=1,N3
      DO 35 M=1,4,3
      YTEMP=SINE*A(L,M,K)+COSINE*A(L,M+1,K)
      A(L,M,K-1)=COSINE*A(L,M,K)-SINE*A(L,M+1,K)
      A(L,M+1,K-1)=COSA*YTEMP+SINA*A(L,M+2,K)
      A(L,M+2,K-1)=-SINA*YTEMP+COSA*A(L,M+2,K)
35    A(L,M/4+7,K-1)=A(L,M/4+7,K)
36    DO 38 K=2,N2
      IPRINT=0
      IF (I.NE.NS) GO TO 37
      IPRINT=KPRINT(1)
      IF (J.NE.NL) GO TO 37
      IPRINT=KPRINT(2)
      IF (K.NE.NP+1) GO TO 37
      IPRINT=KPRINT(3)
37    CALL BULK (K,IND)
      IF (ITEST.NE.KTEST) GO TO 15
      IF (KURE.NE.ILL) GO TO 22
      IF (IND.NE.3) CALL DATOUT (A,K,IND,KAB,0)
```

```
38    CONTINUE
      IF (IND.NE.3) WRITE (NTAPE) ((A(L,K),K=1,8),L=2,N2)
      IF (IND-2) 42,45,39
39    CALL HARNES (QINF,A,NPC)
      DO 41 K=2,4
      N=K
      IF (K.EQ.4) N=1
      DO 41 L=1,N3
      IF (K.EQ.4.AND.(L.EQ.1.OR.L.EQ.N3)) GO TO 41
      DO 40 M=1,4,3
      A(L,M/4+7,K)=A(L,M/4+7,N)
      YTEMP=COSA*A(L,M+1,N)-SINA*A(L,M+2,N)
      A(L,M+2,K)=SINA*A(L,M+1,N)+COSA*A(L,M+2,N)
      A(L,M+1,K)=-SINE*A(L,M,N)+COSINE*YTEMP
40    A(L,M,K)=COSINE*A(L,M,N)+SINE*YTEMP
41    CONTINUE
      GO TO 45
42    LB=2
      IF (K1.NE.0) LB=K1+1
      LE=NPC
      IF (K2.NE.0) LE=K2
      DO 44 L=LB,LE
      ARC=(A(L+1,1)-A(L,1))**2+(A(L+1,2)-A(L,2))**2+(A(L+1,3)-A(L,3))**2
      IF (L.NE.LB) GO TO 43
      BIG=ARC
      SMALL=ARC
43    BIG=AMAX1(BIG,ARC)
44    SMALL=AMIN1(SMALL,ARC)
      IF (2.*SMALL.LE.BIG) MESS=1
45    CONTINUE
      DO 47 J=2,N2
      DO 46 K=1,8
46    A(J,K)=A(J,K,4)
      CALL DATOUT (A,J,3,KAB,0)
47    CONTINUE
      IF (KP4.GT.0.OR.(KP4.EQ.0.AND.I/10*10.EQ.I)) WRITE (IOU3,60) I
      WRITE (NTAPE) ((A(J,L),L=1,8),J=2,N2)
      CALL SETCP(NTAPE,2,A,B,SINE,COSINE,POP,GF,FMSTR,N2)
      CALL AERO (NPC,SINA,COSA)
      REWIND NTAPE
      READ (NTAPE) A(1),A(2)
      Y0=SINE*A(1)+COSINE*A(2)
      IF (KP4.EQ.0.AND.I/10*10.NE.I) GO TO 48
```

```
      WRITE (IOU1,62) I,LINE,ILL,KURE,MESS,INCRES,K1,K2,NPC
      WRITE (IOU1,66) SINE,COSINE,S2,C2,DYC,STEP,Y0,DRAG,ALIFT,TORQUE
      IF (I/KAB*KAB.NE.I.AND.KP4.NE.0) GO TO 49
48    IF (KP4.GE.0) WRITE (IOU6,62) I,LINE,ILL,KURE,MESS,INCRES,K1,K2,
     1NPC
      IF (KP4.LT.0) WRITE (IOU7,56) I,LINE,ILL,KURE,MESS,INCRES,K1,K2,
     1NPC
      IF (KP4.GE.0) WRITE (IOU6,66) SINE,COSINE,S2,C2,DYC,STEP,Y0
     * ,DRAG,ALIFT,TORQUE
      IF (KP4.LT.0) WRITE (IOU7,57) SINE,COSINE,S2,C2,DYC,STEP,Y0
     * ,DRAG,ALIFT,TORQUE
49    IF (KP4.LT.1.AND.I/KAB*KAB.NE.I) GO TO 51
      REWIND NTAPE
      DO 50 J=1,LINE
      READ (NTAPE) ((A(K,L),L=1,8),K=2,N2)
      IND=2
      IF (J.EQ.1) IND=1
      IF (J.EQ.LINE) IND=3
      DO 50 K=2,N2
50    CALL DATOUT (A,K,IND,KAB,1)
      IF (KP4.GT.0.OR.(KP4.EQ.0.AND.I/10*10.EQ.I)) END FILE IOU3
51    CONTINUE
      CALL EXIT
C
52    FORMAT (18A4)
53    FORMAT (F10.5,12I5)
54    FORMAT (1H140X38HNORTHROP SUPERSONIC FLOW FIELD PROGRAM//30X18A4)
55    FORMAT (7F10.5)
56    FORMAT (16I5)
57    FORMAT (4E15.8)
58    FORMAT (1H039X40H 3-DIMENSIONAL CHARACTERISTICS SOLUTION /1H05X,24
     1HFREESTREAM MACH NUMBER =,F7.3,5X25HRATIO OF SPECIFIC HEATS =,F6.3
     2,5X17HANGLE OF ATTACK =,F7.3,8H DEGREES/1H049X21HINITIAL VALUE SUR
     3FACE,T2,6HDRAG =,G14.7/T2,6HLIFT =,G14.7/T2,8HMOMENT =,G14.7)
59    FORMAT (2E15.8,F10.7,E15.8,F12.8,F13.8)
60    FORMAT (1H150X15HDATA SURFACE NO,I3/93H0***TO USE THIS AS AN INITI
     1AL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS
     2)
61    FORMAT (33H0ILLNATURED PSI AND DELTA,CASE NO,I3,42H, ROTATE COORDI
     1NATES AND RECOMPUTE SURFACE,I3)
62    FORMAT (61H0TO USE AS IVS, PUNCH THE 4 CARDS, PLACE BEFORE BODY PT
     * CARDS/1X,9(1H0),10(1H1),10(1H2),10(1H3),10(1H4),10(1H5),10(1H6),
     * 10(1H7),1H8/1X,8(10H1234567890)//1X,14I5)
```

```
63    FORMAT (42H0DATA POINTS WILL BE RE-SPACED FOR SURFACE,I3)
64    FORMAT (13H0WRONG IVS NOI4,4H, NOI4,27H SPECIFIED OR FAIL END FILE
     1)
65    FORMAT (1X/////1X,14I5)
66    FORMAT (1X,4E15.8)
67    FORMAT (1X,2E15.8,F10.7,E15.8,F12.8,F13.8)
68    FORMAT (1H1,30X18A4,/1H05X24HFREESTREAM MACH NUMBER =F7.3,5X25HRAT
     1IO OF SPECIFIC HEATS =F6.3,5X,17HANGLE OF ATTACK =F7.3,8H DEGREES)
      END
```

```
*DECK B
      SUBROUTINE BULK (L,IND)
C     IND =1 FOR BODY, 2 FOR FIELD, 3 FOR SHOCK
C**** TAKE NOTE THAT SINE,COSINE MUST CORRESPOND TO CURRENT  COORDINATES
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      COMMON /STOR/ H(4,8),XA1(50),YA1(50),ZA1(50),UA1(50),VA1(50),WA1(
     150),PA1(50),PPMA1(50)
      DIMENSION XYZ(3,7)
      EQUIVALENCE (XC,XYZ)
1     IF (IPRINT.GE.1) 2,3
2     IF (IND.EQ.1) WRITE (IOU6,14)
      IF (IND.EQ.2) WRITE (IOU6,13)
      IF (IND.EQ.3) WRITE (IOU6,15)
      N=4-MOD(IND,2)
3     CALL PICK (L,IND)
      IF (KURE.NE.ILL) RETURN
C     BACK TO MAIN PROGRAM, ROTATE Y AXIS AWAY FROM FLOW FIELD
      UC=HUB(4)
      VC=HUB(5)
      WC=HUB(6)
      PC=HUB(7)
      PPMC=HUB(8)
      THETAC=TH
      PSIC=PS
      DO 4 I=1,4
4     LOT(I)=0
      DO 11 I=1,25
      IF (IPRINT.GE.1) WRITE (IOU6,16) I
      IF (IND.NE.3) GO TO 5
      Q=SQRT(1.-(PC/PPMC)**GF(7))
      CALL NORM (-UC*Q,QINF-VC*Q,-WC*Q,A,B,C)
      IF (I.EQ.1) CALL NORM (A,B,C,A1,B1,C1)
      CALL NORM (A+A1,B+B1,C+C1,A,B,C)
      IF (IPRINT.GE.1) WRITE (IOU6,21) A,B,C,A1,B1,C1
      A4=-A*B
      B4=1.-B*B
```

```
      C4=-C*B
      BM=COSA*B4-SINA*C4
      AB=(SINA*B4+COSA*C4)/BM
      XC=HUB(1)+DYC*A4/BM
      YC=HUB(2)+DYC*(COSA+SINA*AB)
      ZC=HUB(3)+DYC*(-SINA+COSA*AB)
      GO TO 6
    5 A4=UC+HUB(4)
      B4=VC+HUB(5)
      C4=WC+HUB(6)
      BM=SINE*A4+COSINE*B4
      AB=(COSINE*A4-SINE*B4)/BM
      XC=HUB(1)+DYC*(SINE+COSINE*AB)
      YC=HUB(2)+DYC*(COSINE-SINE*AB)
      ZC=HUB(3)+DYC*C4/BM
    6 IF (IPRINT.GE.1) WRITE (IOU6,21) A4,B4,C4,BM,AB,XC,YC,ZC
      IF (IND.NE.1) GO TO 7
      CALL GMTRY (SINE,COSINE,XC,YC,ZC,A4,B4,C4)
      IF (IPRINT.GE.1) WRITE (IOU6,21) XC,YC,ZC,A4,B4,C4
      CALL TRANS (H,XYZ,A4,B4,C4,1.,1)
    7 CALL BASEPT (IND)
      IF (ITEST.EQ.KTEST.OR.LOT(4).NE.4) GO TO 75
      H(4,1) = XA1(L)
      H(4,2) = YA1(L)
      H(4,3) = ZA1(L)
      CALL NORM(UA1(L),VA1(L),WA1(L),H(4,4),H(4,5),H(4,6))
      H(4,7) = PA1(L)
      H(4,8) = PPMA1(L)
      KTEST = ITEST
      GO TO 7
   75 IF (KURE.NE.ILL.OR.ITEST.NE.KTEST) RETURN
      IF (IPRINT.GE.1) WRITE (IOU6,17) (J,XBS(J),YBS(J),ZBS(J),PBS(J),
     1PPMBS(J),THBS(J),PSBS(J),UBS(J),VBS(J),WBS(J),J=1,N)
      CALL COEFS (IND)
      IF (KURE.NE.ILL) RETURN
      IF (IND.EQ.1) CALL SOL (COF(4),COF(6),COF(5),COF(1),COF(6),COF(9),
     1COF(8),COF(3),0.,0.,1.,1.5707963267949,PC,PSIC,THETAC)
      IF (IND.EQ.2) CALL SOL (COF(4),COF(5),COF(6),COF(1),COF(5),COF(7),
     1COF(8),COF(2),COF(6),COF(8),COF(9),COF(3),PC,THETAC,PSIC)
      IF (IND.EQ.3) CALL COMPAT
      SINTH=SIN(THETAC)
      UC=SINTH*COS(PSIC)
      VC=COS(THETAC)
```

```
      WC=SINTH*SIN(PSIC)
      IF (IND.EQ.1) CALL TRANS (H,XYZ,A4,B4,C4,-1.,1)
      IF (UC*UC+WC*WC.GE.4.E-06) GO TO 8
      ILL=ILL+1
      RETURN
8     IF (IPRINT.GE.1) WRITE (IOU6,18) I,XC,YC,ZC,PC,PPMC,UC,VC,WC,
     1THETAC,PSIC
      IF (L.NE.2.AND.L.NE.NPC+1) GO TO 9
      UT=SINE*UC+COSINE*VC
      IF (IND.NE.3) CALL NORM (SINE*UT,COSINE*UT,WC,UC,VC,WC)
      IF (IND.EQ.3) CALL NORM (0.,VC,WC,UC,VC,WC)
9     IF (I.EQ.1) GO TO 10
      IF (ABS(THETAC-THTEMP).GT.DTHETA) GO TO 10
      IF (ABS(PSIC-PSTEMP).GT.DPSI) GO TO 10
      IF (ABS((PC-PTEMP)/PC).LE.DP) GO TO 12
10    THTEMP=THETAC
      PSTEMP=PSIC
11    PTEMP=PC
      WRITE (IOU6,19)
      CALL EXIT
12    L1=L-1
      IF (IPRINT.GE.1) WRITE (IOU6,20) L1
      Q=SQRT(1.-(PC/PPMC)**GF(7))
      XA1(L)=XC
      YA1(L)=YC
      ZA1(L)=ZC
      UA1(L)=Q*UC
      VA1(L)=Q*VC
      WA1(L)=Q*WC
      PA1(L)=PC
      PPMA1(L)=PPMC
      RETURN
C
13    FORMAT (18H0***** FIELD *****)
14    FORMAT (17H0***** BODY *****)
15    FORMAT (18H0***** SHOCK *****)
16    FORMAT (1H0,20X,32HBASE POINT PROPERTIES, ITERATION,I3)
17    FORMAT (15H0I,X,Y,Z,P,PPM=,I5,5E18.8/17H THETA,PSI,U,V,W=,3X5E18.8
     1)
18    FORMAT (41H0NEW BULK POINT PROPERTIES, ITERATION NO.,I3/1H 21X,2HX
     1C,18X,2HYC,18X,2HZC,18X,2HPC,17X,4HPPMC/1H ,10X,5E20.8/1H ,21X,2HU
     2C,18X,2HVC,18X,2HWC,16X,6HTHETAC,15X,4HPSIC/1H ,10X,5E20.8)
19    FORMAT (54H0BULK POINT HAS FAILED TO CONVERGE AFTER 25 ITERATIONS)
```

```
20    FORMAT (24H0********* BULK POINT NO.,I3,23H HAS CONVERGED ********)    B 120
21    FORMAT (8E15.7)                                                        B 121
      END                                                                    B 122
```

```
*DECK C
      SUBROUTINE PICK (L,IND)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /ESFNS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /STOR/ H(4,8),ALMUS(50,8,4)
1     LP=L+1
      LM=L-1
      N=2
      IF (IND.EQ.2) N=4
      DO 2 I=1,8
      HUB(I)=ALMUS(L,I,3)
      IF (IND.NE.3) H(2,I)=ALMUS(L,I,4)
      IF (IND.NE.1) H(N,I)=ALMUS(L,I,2)
      H(1,I)=ALMUS(LM,I,3)
2     H(3,I)=ALMUS(LP,I,3)
      CALL NORM (HUB(4),HUB(5),HUB(6),HUB(4),HUB(5),HUB(6))
      IF (HUB(4)**2+HUB(6)**2.GT.4.E-06) GO TO 3
      ILL=ILL+1
      RETURN
3     TH=ACOS(HUB(5))
      PS=ATAN2(HUB(6),HUB(4))
      IF (IND.NE.2) N=3
      DO 4 I=1,N
4     CALL NORM (H(I,4),H(I,5),H(I,6),H(I,4),H(I,5),H(I,6))
      IF (IPRINT.GE.2) WRITE (IOU6,5) HUB,((H(I,J),J=1,8),I=1,N),TH,PS
      RETURN
C
5     FORMAT (17H0***** PICK ****/(8E15.7))
      END
```

```
*DECK D
      SUBROUTINE BASEPT (IND)                                           D   1
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST,KPRINT(3)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI            D   3
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7)                          D   4
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE                        D   5
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH, D   6
     1PS,HUB(8)                                                         D   7
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),   D   8
     1PPMBS(4),THBS(4),PSBS(4)                                          D   9
      COMMON /STOR/ XH(4),YH(4),ZH(4),UH(4),VH(4),WH(4),PH(4),PPMH(4)   D  10
      IF (IPRINT.GE.2) WRITE (IOU6,13)                                  D  11
      XD=XC-HUB(1)                                                      D  12
      YD=YC-HUB(2)                                                      D  13
      ZD=ZC-HUB(3)                                                      D  14
      B2=XD*XD+YD*YD+ZD*ZD                                              D  15
      ROOT = GF(2)/((PPMC/PC)**GF(7)-1.0)
      IF(ROOT.GT.0.) GO TO 20
      WRITE(IOU6,600) PPMC,PC
      CALL EXIT
  600 FORMAT(// 24H FLOW SUBSONIC, PPMC,PC= ,2G15.7)
   20 AMU4=ASIN(SQRT(ROOT))
      IF (IPRINT.GE.2) WRITE (IOU6,14) XD,YD,ZD,AMU4                    D  17
      LID=3                                                             D  18
      IF (IND.EQ.2) LID=4                                               D  19
      N = 1                                                             D 194
      IF (LOT(4).EQ.4) N=4                                              D 195
      DO 8 I=N,LID                                                      D  20
      XD1=XH(I)-HUB(1)                                                  D  21
      YD1=YH(I)-HUB(2)                                                  D  22
      ZD1=ZH(I)-HUB(3)                                                  D  23
      IF (IPRINT.GE.2) WRITE (IOU6,15) I                                D  24
      A2=XD1*XD1+YD1*YD1+ZD1*ZD1                                        D  25
      AB=XD*XD1+YD*YD1+ZD*ZD1                                           D  26
      IF (IPRINT.GE.2) WRITE (IOU6,14) XD1,YD1,ZD1                      D  27
      DO 5 J=1,50                                                       D  28
      IF (J.EQ.1) FL=.5                                                 D  29
      IF (J.EQ.1) GO TO 4                                               D  30
      COS2=COS(.5*(AMU0+AMU4))**2                                       D  31
      CALL NORM (.5*(UBS(I)+UC),.5*(VBS(I)+VC),.5*(WBS(I)+WC),U,V,W)    D  32
      BV=XD*U+YD*V+ZD*W                                                 D  33
      AV=XD1*U+YD1*V+ZD1*W                                              D  34
      FL=0.                                                             D  35
```

```
      A=A2*COS2-AV*AV                                                   D  36
      B=AB*COS2-AV*BV                                                   D  37
      C=B2*COS2-BV*BV                                                   D  38
      TERM=B*B-A*C                                                      D  39
      IF (TERM.LT.0.) GO TO 9                                           D  40
      IF (A.LE.0..AND.B.GT.0.) GO TO 9                                  D  41
      IF (C) 1,3,1                                                      D  42
1     FL=C/(B-SQRT(TERM))                                               D  43
      IF(IPRINT.GE.2)WRITE(IOU6,14)COS2,BV,AV,U,V,W,A,B,C,TERM,FL,A2,AB
      IF (LOT(I).NE.0.OR.(FL.GE.0..AND.FL.LE..7)) GO TO 2               D  44
      IF((I.NE.4).OR.(KPRINT(3).GT.0.)) WRITE(IOU6,10)I,FL
      LOT(I)=I                                                          D  46
      IF (I.NE.4) STEP=-.1
2     IF (FL.GE.0..AND.FL.LE.1.) GO TO 3                                D  48
      IF((I.NE.4).OR.(KPRINT(3).GT.0)) WRITE(IOU6,16)I,FL
      IF (N.EQ.4) GO TO 3
200   ITEST = ITEST+1
      RETURN                                                            D  51
3     XBS(I)=HUB(1)+FL*XD1                                              D  52
      YBS(I)=HUB(2)+FL*YD1                                              D  53
      ZBS(I)=HUB(3)+FL*ZD1                                              D  54
4     UBS(I)=HUB(4)+FL*(UH(I)-HUB(4))                                   D  55
      VBS(I)=HUB(5)+FL*(VH(I)-HUB(5))                                   D  56
      WBS(I)=HUB(6)+FL*(WH(I)-HUB(6))                                   D  57
      PBS(I)=HUB(7)+FL*(PH(I)-HUB(7))                                   D  58
      PPMBS(I)=HUB(8)+FL*(PPMH(I)-HUB(8))                               D  59
      CALL NORM (UBS(I),VBS(I),WBS(I),UBS(I),VBS(I),WBS(I))             D  60
      AMU=ASIN(SQRT(GF(2)/((PPMBS(I)/PBS(I))**GF(7)-1.)))               D  61
      IF (IPRINT.GE.2) WRITE (IOU6,14) U,V,W,A,B,C,TERM,FL,AMU
     1   ,PBS(I),PPMBS(I),HUB(7),WH(I),HUB(8),PPMH(I)
      IF (J.EQ.1) GO TO 5
      IF (FL.GT.1.) GO TO 6
      IF (ABS(AMU-AMU0).LE.EPSLN) GO TO 6                               D  64
5     AMU0=AMU                                                          D  65
      WRITE (IOU6,12) I                                                 D  66
      CALL EXIT                                                         D  67
6     IF (UBS(I)**2+WBS(I)**2.GT.4.E-06) GO TO 7                        D  68
      ILL=ILL+1                                                         D  69
      RETURN                                                            D  70
7     THBS(I)=ACOS(VBS(I))                                              D  71
      PSBS(I)=ATAN2(WBS(I),UBS(I))                                      D  72
8     CONTINUE                                                          D  73
      IF (N.EQ.4) LOT(4) = 0
```

```
      RETURN                                                            D  74
9     WRITE (IOU6,11) I,A,B,C,TERM                                      D  75
      IF (I.NE.4.OR.N.EQ.4) CALL EXIT
      LOT(4) = 4
      GO TO 200
C                                                                       D  77
10    FORMAT (11H BASE POINT,I2,13H VIOLATES CFL,F10.5)                 D  78
11    FORMAT (31H0INTERSEXION FAILURE, BASEPT NO,I2,4E18.8)             D  79
12    FORMAT (43H0MACH CONE CONVERGENCE FAILURE IN BASEPT NO,I2)        D  80
13    FORMAT (19H0***** BASEPT *****)                                   D  81
14    FORMAT (8E15.7)                                                   D  82
15    FORMAT (14H0BASE POINT NO,I2)                                     D  83
16    FORMAT (8H BASE PT,I2,13H EXTRAPOLATES,F10.5,16H, STEP DECREASED) D  84
      END                                                               D  85
```

```
*DECK E
      SUBROUTINE COEFS (IND)
C     IND = 1 FOR BODY, 2 FOR FIELD, 3 FOR SHOCK
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      DIMENSION DX(4),DY(4),DZ(4),DTH(4),DPS(4),A(4),B(4),C(4),D(4)
1     IF (IPRINT.GE.2) WRITE (IOU6,8)
      N=4
      IF (IND.EQ.2) GO TO 2
      N=3
      DX(4)=XC-HUB(1)
      DY(4)=YC-HUB(2)
      DZ(4)=ZC-HUB(3)
      DTH(4)=THETAC-TH
      DPS(4)=PSIC-PS
2     AMUC=ASIN(SQRT(GF(2)/((PPMC/PC)**GF(7)-1.)))
      DO 3 I=1,N
      DX(I)=XC-XBS(I)
      DY(I)=YC-YBS(I)
      DZ(I)=ZC-ZBS(I)
      DTH(I)=THETAC-THBS(I)
      DPS(I)=PSIC-PSBS(I)
      IF (IPRINT.GE.2) WRITE (IOU6,9) DX(I),DY(I),DZ(I),DTH(I),DPS(I)
3     CONTINUE
      DO 5 I=1,N
      CALL NORM (.5*(UBS(I)+UC),.5*(VBS(I)+VC),.5*(WBS(I)+WC),UNT,VNT,
     1WNT)
      IF (IPRINT.GE.2) WRITE (IOU6,9) UNT,VNT,WNT
      J=MOD(I,4)+1
      K=MOD(I-5,4)+4
      CALL SOL (DX(I),DY(I),DZ(I),DTH(I),DX(J),DY(J),DZ(J),DTH(J),DX(K),
     1DY(K),DZ(K),DTH(K),DTHDX,DTHDY,DTHDZ)
      CALL SOL (DX(I),DY(I),DZ(I),DPS(I),DX(J),DY(J),DZ(J),DPS(J),DX(K),
     1DY(K),DZ(K),DPS(K),DPSDX,DPSDY,DPSDZ)
      CALL NORM (VNT*DZ(I)-WNT*DY(I),WNT*DX(I)-UNT*DZ(I),UNT*DY(I)-VNT*
     1DX(I),DXDN,DYDN,DZDN)
      DTHDN=DTHDX*DXDN+DTHDY*DYDN+DTHDZ*DZDN
      DPSDN=DPSDX*DXDN+DPSDY*DYDN+DPSDZ*DZDN
```

```
      IF (IPRINT.GE.2) WRITE (IOU6,9) DXDN,DYDN,DZDN,DTHDN,DPSDN
      VCHEK=SQRT(UNT*UNT+WNT*WNT)
      IF (VCHEK.GT.2.E-3) GO TO 4
      ILL=ILL+1
      RETURN
4     COSD1=(UNT*DZDN-WNT*DXDN)/VCHEK
      IF (ABS(COSD1).GT.1.) COSD1=SIGN(1.,COSD1)
      UZWX=UNT*DZ(I)-WNT*DX(I)
      SIND1=SIGN(SQRT(1.-COSD1*COSD1),UZWX)
      AMUAV=.5*(AMUC+ASIN(SQRT(GF(2)/((PPMBS(I)/PBS(I))**GF(7)-1.))))
      SINMU=SIN(AMUAV)
      SINTH=SIN(.5*(THETAC+THBS(I)))
      DL=SQRT(DX(I)*DX(I)+DY(I)*DY(I)+DZ(I)*DZ(I))
      IF (IPRINT.GE.2) WRITE (IOU6,9) AMUAV,SINMU,SINTH,SIND1,COSD1,DL
      A(I)=SINMU*COS(AMUAV)/(GF(1)*.5*(PC+PBS(I)))
      B(I)=-COSD1
      C(I)=SINTH*SIND1
      D(I)=A(I)*PBS(I)+B(I)*THBS(I)+C(I)*PSBS(I)-DL*SINMU*(SIND1*DTHDN+
     1SINTH*COSD1*DPSDN)
      IF (IPRINT.GE.2) WRITE (IOU6,9) A(I),B(I),C(I),D(I)
5     CONTINUE
      DO 6 I=1,9
6     COF(I)=0.
      DO 7 I=1,N
      COF(1)=COF(1)+A(I)*D(I)
      COF(2)=COF(2)+B(I)*D(I)
      COF(3)=COF(3)+C(I)*D(I)
      COF(4)=COF(4)+A(I)*A(I)
      COF(5)=COF(5)+A(I)*B(I)
      COF(6)=COF(6)+A(I)*C(I)
      COF(7)=COF(7)+B(I)*B(I)
      COF(8)=COF(8)+B(I)*C(I)
7     COF(9)=COF(9)+C(I)*C(I)
      IF (IPRINT.GE.2) WRITE (IOU6,9) (COF(I),I=1,9)
      RETURN
C
8     FORMAT (18H0***** COEFS *****)
9     FORMAT (8E15.7)
      END
```

```
*DECK F
      SUBROUTINE COMPAT
C**** Y-AXIS MUST ALIGN WITH FREE STREAM, ROTATE AXES IF NEEDED
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC
      IF (IPRINT.GE.2) WRITE (IOU6,4)
      GM2=GF(1)*FMSTR*FMSTR
      DO 2 I=1,20
      XI=PC*POP
      D1=XI-1.
      D2=GM2-D1
      D4=XI+GF(4)
      D3=GM2/GF(3)-D4
      THETAC=ATAN(SQRT(D3/D4)*D1/D2)
      PSIC=(COF(3)-COF(8)*THETAC-COF(6)*PC)/COF(9)
      SN2T=.5*SIN(2.*THETAC)
      DTHDP=SN2T*(1./D1+1./D2-.5/D3-.5/D4)*POP
      DPSDP=-(COF(8)*DTHDP+COF(6))/COF(9)
      D2TDP2=COS(2.*THETAC)/SN2T*DTHDP*DTHDP+POP**2*SN2T*(-1./D1**2+1./
     1D2**2-.5/D3**2+.5/D4**2)
      C=COF(5)*PC+COF(7)*THETAC+COF(8)*PSIC-COF(2)
      H=C*DTHDP+COF(4)*PC+COF(5)*THETAC+COF(6)*PSIC-COF(1)
      HP=C*D2TDP2+(COF(7)*DTHDP+COF(8)*DPSDP+2.*COF(5))*DTHDP+COF(6)*
     1DPSDP+COF(4)
      PC=PC-H/HP
      IF (IPRINT.GE.2) WRITE (IOU6,5) PC,THETAC,PSIC,XI,D1,D2,D3,D4,SN2T
     1,DTHDP,DPSDP,D2TDP2,C,H,HP
      IF (I.EQ.1) GO TO 1
      IF (ABS((PC-PCN)/PC).GT.DP) GO TO 1
      IF (ABS(THETAC-THCN).GT.DTHETA) GO TO 1
      IF (ABS(PSIC-PSCN).LE.DPSI) GO TO 3
1     PCN=PC
      THCN=THETAC
2     PSCN=PSIC
      WRITE (IOU6,6)
      CALL EXIT
3     PPMC=(D4/(GF(4)*XI+1.))**GF(6)/XI**GF(5)
      RETURN
C
4     FORMAT (19H0***** COMPAT *****)
5     FORMAT (8E15.7)
6     FORMAT (30H0CONVERGENCE FAILURE IN COMPAT)
```

F 43

END

```
*DECK G
      SUBROUTINE GMTRY (SINE,COSINE,XC,YC,ZC,A4,B4,C4)                  G0000
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI            G0010
      COMMON/GEOM/ YMAX,YSTOP,INDEX                                     G0020
      X1=COSINE*XC-SINE*YC                                              G0040
      YC=SINE*XC+COSINE*YC                                              G0050
      TA=COSINE*A4-SINE*B4                                              G0060
      TB=SINE*A4+COSINE*B4                                              G0070
      XC=X1                                                             G0080
      Z1=ZC                                                             G0090
      IND=0                                                             G0100
      DO 10 J=1,5                                                       G0110
      INDEX=0                                                           G0120
      IN=IND                                                            G0130
      T=0.                                                              G0140
      DO 1 I=1,20                                                       G0150
      CALL FIGURE(XC,YC,ZC,F,A,B,C)                                     G0160
      IF(INDEX.EQ.IND) GO TO 2                                          G0170
      IF(I.GT.1) GO TO 9                                                G0180
      B1=TA*B-TB*A                                                      G0190
      B3=C4*B-TB*C                                                      G0200
    9 T=T-F/(B1*A+B3*C)                                                 G0210
      XC=X1+T*B1                                                        G0220
      ZC=Z1+T*B3                                                        G0230
      IF (IPRINT.GE.2) WRITE (IOU6,3)                                   G0235
      IF(IPRINT.GE.2)WRITE(IOU6,4)T,XC,YC,ZC,F,A,B,C,INDEX,IND,J        G0240
      IF (ABS(F).LE.1.E-6) GO TO 10                                     G0250
    1 CONTINUE                                                          G0260
      WRITE (IOU6,5)                                                    G0270
      CALL EXIT                                                         G0280
   10 IND=INDEX                                                         G0290
      WRITE(IOU6,6) INDEX,IN                                            G0300
    2 CALL ROTATE (XC,YC,SINE,COSINE)                                   G0310
      CALL NORM (COSINE*A+SINE*B,-SINE*A+COSINE*B,C ,A4,B4,C4)          G0320
      RETURN                                                            G0330
C                                                                       G0340
3     FORMAT(25(2H *),6H GMTRY,25(2H *))                                G0350
4     FORMAT (8G15.7,3I3)                                               G0360
5     FORMAT (29H0CONVERGENCE FAILURE IN GMTRY)                         G0370
6     FORMAT (40H0BODY POINT OSCILLATING BETWEEN SECTIONS,I2,4H AND,I2) G0380
      END                                                               G0390
```

```
*DECK H
      SUBROUTINE FIGURE (X,Y,Z,F,A,B,C)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /GEOM/ YMAX,YSTOP,INDEX
      DIMENSION ZX(13),ZXP(13),P(13),Q(13),R(13),S(13),T(13),SG(13)
      IF(Y.GT.YSTOP+.0001.AND.ABS(YSTOP).GT.EPSLN) GO TO 7
      DO 2 I=1,50
      IF (Y.LE.YMAX) GO TO 3
      M=7
      READ (IOU5,8) YMAX,N
      IF(EOF(IOU5)) 7,1
1     WRITE (IOU6,12) YMAX,N
      DO 2 K=1,N
      READ (IOU5,10) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
      IF(L.GT.7) M=13
      WRITE (IOU6,13) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
    2 CONTINUE
    3 DO 4 I=1,M
      CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
    4 CONTINUE
      IF(INDEX.GT.0)GO TO (101,201,301,401,501,601),INDEX
      IF(M.EQ.7) GO TO 50
      X1=(ZX(12)-ZX(13))*
     *                SQRT(1.-((ZX(8)-ZX(11))/(ZX(10)-ZX(11)))**2)+ZX(13)
      IF((Z.GT.ZX(2)).AND.(X.LT.ZX(13))) GO TO 600
      IF((Z.GT.ZX(8)).AND.(X.LT.X1)) GO TO 100
      IF((ZX(8)-ZX(9)).LE.0.) GO TO 50
      IF(Z.GT.ZX(9)) GO TO 200
   50 IF(Z.GT.ZX(2)) GO TO 300
      IF(X.GT.ZX(6)) GO TO 400
C****LOWER FLAT SECTION
      INDEX=5
  501 A=0.0
      C=-1.0
      B=ZXP(3)
      F=-(Z-ZX(3))
      GO TO 6
C****CANOPY
  100 INDEX=1
  101 CALL ELIPSE(X,Z,ZX,ZXP,10,11,13,12,A,B,C,F)
      GO TO 6
C****CUBIC SECTION
  200 INDEX=2
```

```
  201 CALL CUBIC(X,Z,ZX,ZXP,A,B,C,F)                                     H
      GO TO 6                                                            H
C****UPPER ELLIPSE SECTION                                               H
  300 INDEX=3                                                            H
  301 CALL ELIPSE(X,Z,ZX,ZXP,1,2,4,5,A,B,C,F)                            H
      GO TO 6                                                            H
C****LOWER CONIC SECTION                                                 H
  400 INDEX=4                                                            H
  401 CALL KONIC(X,Z,ZX,ZXP,6,3,5,2,7,A,B,C,F)                           H
      GO TO 6                                                            H
C****CANOPY FLAT SECTION                                                 H
  600 INDEX=6                                                            H
  601 A=0.0                                                              H
      C=-1.0                                                             H
      B=ZXP(10)                                                          H
      F=-(Z-ZX(10))                                                      H
    6 IF (IPRINT.GE.2) WRITE(IOU6,9) F,A,B,C,INDEX                       H
      RETURN                                                             H
7     WRITE (IOU6,11)                                                    H
      CALL EXIT                                                          H
C                                                                        H
8     FORMAT (F10.5,I5)                                                  H
9     FORMAT(19H0***** FIGURE *****/4G14.7,21H BODY SECTION INDEX =,I2)  H
10    FORMAT (I2,5E15.8,F2.0)                                            H
11    FORMAT(1H0,60(2H *)/1H0,20X,35H$$$$$ HALLELUJAH, YOU HAVE GONE ALLH
     1, 35H THE WAY--FEELING NICE, I BET $$$$$/1H0,60(2H *))             H
12    FORMAT (35H0GEOMETRIC COEFFICIENTS FOR Y UP TO,G15.7,I5,11H NEW CUH
     1RVES)                                                              H
13    FORMAT (9H0CURVE NO,I3,5H, P =E16.8,5H  Q =E16.8,5H  R =E16.8,5H   H
     1S =E16.8,5H  T =E16.8,6H  SG =F4.1)                                H
      END                                                                H
```

```
*DECK I
      SUBROUTINE TRANS (H,XYZ,A4,B4,C4,TO,IN)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      DIMENSION H(4,8),XYZ(3,7)
      IF (A4*A4+C4*C4.LE.1.E-14) RETURN
      COSX=-SQRT(1.-C4*C4)
      SRT=SQRT(A4*A4+B4*B4)
      IF (SRT.GT.1.E-7) GO TO 1
      COSZ=0.
      SINZ=1.*TO
      GO TO 2
1     COSZ=-B4/SRT
      SINZ=TO*A4/SRT
2     DO 5 J=1,7
      IF (IN.EQ.0.AND.J.GT.4) GO TO 5
      IF (IN.NE.0.AND.J.GT.3) GO TO 4
      DO 3 K=1,4,3
      IF (TO.GT.0.) CALL ROTATE (H(J,K),H(J,K+1),SINZ,COSZ)
      CALL ROTATE (H(J,K+1),H(J,K+2),C4*TO,COSX)
      IF (TO.LT.0.) CALL ROTATE (H(J,K),H(J,K+1),SINZ,COSZ)
3     CONTINUE
      IF (IN.EQ.0) GO TO 5
4     IF (J.GT.2.AND.J.LT.6) GO TO 5
      IF (TO.GT.0.) CALL ROTATE (XYZ(1,J),XYZ(2,J),SINZ,COSZ)
      CALL ROTATE (XYZ(2,J),XYZ(3,J),C4*TO,COSX)
      IF (TO.LT.0.) CALL ROTATE (XYZ(1,J),XYZ(2,J),SINZ,COSZ)
5     CONTINUE
      IF (IPRINT.GE.2) WRITE (IOU6,6) H,XYZ
      IF (TO.LT.0..OR.IN.EQ.0) RETURN
      XYZ(3,3)=ACOS(XYZ(2,2))
      XYZ(1,4)=ATAN2(XYZ(3,2),XYZ(1,2))
      XYZ(2,5)=ACOS(XYZ(2,7))
      XYZ(3,5)=ATAN2(XYZ(3,7),XYZ(1,7))
      RETURN
C
6     FORMAT (18H0***** TRANS *****/(8E15.7))
      END
```

```
*DECK J
      SUBROUTINE ROTATE (X,Y,SINE,COSINE)
      TEMP=X
      X=COSINE*TEMP+SINE*Y
      Y=-SINE*TEMP+COSINE*Y
      RETURN
      END
```

```
*DECK K
      SUBROUTINE SOL (A11,A12,A13,D1,A21,A22,A23,D2,A31,A32,A33,D3,X,Y,Z
     1)
      IF (ABS(A11).LE.1.E-7) GO TO 1
      B12=A12/A11
      B13=A13/A11
      C1=D1/A11
      B22=A22-B12*A21
      IF (ABS(B22).LE.1.E-7) GO TO 1
      B32=A32-B12*A31
      B23=(A23-B13*A21)/B22
      C2=(D2-A21*C1)/B22
      B33=(A33-A31*B13-B23*B32)
      IF (ABS(B33).LE.1.E-7) GO TO 1
      Z=(D3-C1*A31-C2*B32)/B33
      Y=C2-B23*Z
      X=C1-B12*Y-B13*Z
      RETURN
1     D=A11*(A22*A33-A23*A32)+A21*(A13*A32-A12*A33)+A31*(A12*A23-A13*A22
     1)
      X=(A13*(D2*A32-A22*D3)+A23*(A12*D3-D1*A32)+A33*(D1*A22-A12*D2))/D
      Y=(A11*(D2*A33-A23*D3)+A21*(A13*D3-D1*A33)+A31*(D1*A23-A13*D2))/D
      Z=(A12*(D2*A31-A21*D3)+A22*(A11*D3-D1*A31)+A32*(D1*A21-A11*D2))/D
      RETURN
      END
```

```
*DECK L
      SUBROUTINE HARNES (QINF,A,N)
      DIMENSION A(50,8,4)
      DO 1 L=3,N
      VN=SQRT(A(L,4)**2+(QINF-A(L,5))**2+A(L,6)**2)
      CALL NORM (-A(L,4),QINF-A(L,5),-A(L,6),A0,B0,C0)
      A1=(A(L+1,3)-A(L,3))**2+(A(L+1,1)-A(L,1))**2
      C1=(A(L,3)-A(L-1,3))**2+(A(L,1)-A(L-1,1))**2
      CALL NORM (C1*(A(L+1,3)-A(L,3))+A1*(A(L,3)-A(L-1,3)),0.,C1*(A(L+1,
     11)-A(L,1))+A1*(A(L,1)-A(L-1,1)),A1,B1,C1)
      B=SQRT(1.-B0*B0)
      CALL NORM (SIGN(A1*B,A0)+A0,0.,SIGN(C1*B,C0)+C0,A1,B1,C1)
      A(L,4)=-A1*VN*B
1     A(L,6)=-C1*VN*B
      RETURN
      END
```

```
*DECK M
      SUBROUTINE MIRROR (A,SIN2A,COS2A,N)
      DIMENSION A(50,8)
      I=1
      J=3
      DO 2 L=1,2
      DO 1 K=1,4,3
      A(I,K)=-COS2A*A(J,K)+SIN2A*A(J,K+1)
      A(I,K+1)=SIN2A*A(J,K)+COS2A*A(J,K+1)
      A(I,K+2)=A(J,K+2)
1     A(I,K/4+7)=A(J,K/4+7)
      I=N+L
2     J=N-L
      RETURN
      END
```

```
*DECK N
      SUBROUTINE NORM (X,Y,Z,A,B,C)
      T=SQRT(X*X+Y*Y+Z*Z)
      A=X/T
      B=Y/T
      C=Z/T
      RETURN
      END
```

```
*DECK O
      SUBROUTINE DATOUT (A,I,IND,KAB,IO)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE,N,Y,IS
      DIMENSION A(50,8)
      IF (N.EQ.IS.AND.IO.NE.0) RETURN
      IF (KP4.LE.0.AND.(IO.EQ.1.AND.N/KAB*KAB.NE.N)) RETURN
      IF ((KP4.GT.0.AND.IO.EQ.0).AND.(N/KAB*KAB.NE.N.AND.(IND.NE.1.OR.(
     1I.NE.2.AND.I.NE.NPC+1)))) RETURN
      RAD=57.2957795130823
      P=A(I,7)*POP
      P1=A(I,7)/A(I,8)
      CP=2.*(P-1.)/(GF(1)*FMSTR*FMSTR)
      XM=SQRT((1./P1**GF(7)-1.)/GF(2))
      X=COSINE*A(I,1)-SINE*A(I,2)
      U=COSINE*A(I,4)-SINE*A(I,5)
      V=SINE*A(I,4)+COSINE*A(I,5)
      OW=ATAN(U/V)*RAD
      UW=ATAN(A(I,6)/V)*RAD
      PS=0.
      TH=0.
      IF (ABS(A(I,6)).LE.1.E-07.AND.ABS(U).LE.1.E-07) GO TO 1
      PS=ATAN2(A(I,6),U)*RAD
      TH=ACOS(V/SQRT(U*U+V*V+A(I,6)*A(I,6)))*RAD
1     IF (IO.EQ.0.OR.KP4.GE.0) GO TO 2
      WRITE (IOU7,12) A(I,3),X,XM,A(I,8),TH,PS
      RETURN
2     IOU=IOU6
      IF (KP4.GE.0.AND.IO.EQ.1) IOU=IOU3
      IF (I.GT.2) GO TO 3
      Y=SINE*A(I,1)+COSINE*A(I,2)
      IF (IND.EQ.1) WRITE (IOU,11) N,Y
      IF (KP4.GT.1.AND.(IO.EQ.0.AND.N/KAB*KAB.NE.N)) GO TO 4
      IF (IND.EQ.1) WRITE (IOU,6)
      IF (IND.EQ.2) WRITE (IOU,7)
      IF (IND.EQ.3) WRITE (IOU,8)
      WRITE (IOU,9)
3     IF (KP4.GT.1.AND.(IO.EQ.0.AND.N/KAB*KAB.NE.N)) GO TO 5
      WRITE (IOU,10) A(I,3),X,XM,A(I,8),TH,PS,CP,P,A(I,7),OW,UW
      RETURN
4     WRITE (IOU6,13) CP
5     IF (I.EQ.NPC+1) WRITE (IOU6,14) CP
```

```
      RETURN
C
6     FORMAT (1H054X11HBODY POINTS)
7     FORMAT (1H054X12HFIELD POINTS)
8     FORMAT (1H054X12HSHOCK POINTS)
9     FORMAT (80H ------Z-------  ------X------  MACH NO.  ----PT/PT0---
     1- THETA, DEG  --PSI,DEG--,52H  -----CP-----  ----P/P0----  ---P/PT0-
     2--OUTWASHUPWASH)
10    FORMAT (1H 2E15.8,F10.7,E15.8,F12.8,F13.8,3E13.5,2F6.2)
11    FORMAT (11H0SURFACE NO,I3,14H, STATION Y = E15.8)
12    FORMAT (2E15.8,F10.7,E15.8,F12.8,F13.8)
13    FORMAT (1H+,44X,22HLOWER CENTERLINE CP = E15.8)
14    FORMAT (1H+,83X,22HUPPER CENTERLINE CP = E15.8)
      END
```

```
*DECK P
      SUBROUTINE TIDYUP (SINE,COSINE,A,NP,NPP,AM)
      COMMON /MESH/ K1,K2
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      DIMENSION A(50,8,4)
      DO 1 I=1,NP
      CALL ROTATE (A(I,1),A(I,2),-SINE,COSINE)
1     CALL ROTATE (A(I,4),A(I,5),-SINE,COSINE)
      A(1,1,4)=0.
      DO 2 I=2,NP
      IF (A(I,3).LT.AM) J=I
2     A(I,1,4)=A(I-1,1,4)+SQRT((A(I,1)-A(I-1,1))**2+(A(I,2)-A(I-1,2))**2
     1+(A(I,3)-A(I-1,3))**2)
      TM=(A(J+1,1,4)*(AM-A(J,3))+A(J,1,4)*(A(J+1,3)-AM))/(A(J+1,3)-A(J,3
     1))
      NT=2*NPP-2+K1-K2
      N=0
      DO 3 M=2,NPP
      N=N+1
      IF (M.LE.K1.OR.M.GT.K2) N=N+1
      IF (N.GE.NT/2) GO TO 4
3     CONTINUE
4     DT1=(A(NP,1,4)-TM)/FLOAT(NT-N)
      DO 5 I=1,8
      DT=TM/FLOAT(N)
      L=1
      IF (I.EQ.1.OR.I.EQ.4) L=2
      CALL KURFIT (A(1,1,4),A(1,I),A(1,1,3),NP,0.,0.,L,L)
      PT=-DT-DT
      DO 5 J=1,NPP
      IF (J.GT.M) DT=DT1
      PT=PT+DT
      IF (J.LE.K1.OR.J.GT.K2) PT=PT+DT
      CALL KURVE (A(1,1,3),A(1,1,4),A(1,I),PT,A(J,I,2),NP,0)
5     CONTINUE
      IF (IPRINT.GE.2) WRITE (IOU6,7) ((A(I,J,2),J=1,3),I=1,NPP)
      DO 6 I=1,NPP
      CALL ROTATE (A(I,1,2),A(I,2,2),SINE,COSINE)
6     CALL ROTATE (A(I,4,2),A(I,5,2),SINE,COSINE)
      RETURN
C
7     FORMAT (19H0***** TIDYUP *****/(9E14.6))
      END
```

```
*DECK Q
      SUBROUTINE REFORM (A,LTAPE,NTAPE,NPC,LINE,NLINE)
      DIMENSION A(50,8,4),X(3)
      REWIND LTAPE
      ALINE=NLINE-1
      DO 8 I=1,NPC
      REWIND NTAPE
      DO 4 J=1,LINE
      READ (NTAPE) ((A(L,K,2),K=1,8),L=1,NPC)
      IF (I.NE.1.OR.J.NE.1) GO TO 2
      DO 1 K=1,3
1     X(K)=.5*(A(1,K,2)+A(NPC,K,2))
2     DO 3 K=1,8
3     A(J,K)=A(I,K,2)
4     A(J,8)=ALOG10(-ALOG10(A(J,8)))
      A(1,1,4)=SQRT((A(1,1)-X(1))**2+(A(1,2)-X(2))**2+(A(1,3)-X(3))**2)
      DO 5 J=2,LINE
5     A(J,1,4)=A(J-1,1,4)+SQRT((A(J,1)-A(J-1,1))**2+(A(J,2)-A(J-1,2))**2
     1+(A(J,3)-A(J-1,3))**2)
      DA=(A(LINE,1,4)-A(1,1,4))/ALINE
      DO 6 K=1,8
      CALL KURFIT (A(1,1,4),A(1,K),A(1,1,3),LINE,0.,0.,2,2)
      DO 6 J=1,NLINE
      PA=A(1,1,4)+DA*FLOAT(J-1)
6     CALL KURVE (A(1,1,3),A(1,1,4),A(1,K),PA,A(J,K,2),LINE,K)
      DO 7 J=1,NLINE
7     A(J,8,2)=10.**(-10.**A(J,8,2))
8     WRITE (LTAPE) ((A(M,K,2),K=1,8),M=1,NLINE)
      REWIND NTAPE
      DO 10 J=1,NLINE
      REWIND LTAPE
      DO 9 I=1,NPC
      READ (LTAPE) ((A(M,K,2),K=1,8),M=1,NLINE)
      DO 9 K=1,8
9     A(I,K)=A(J,K,2)
10    WRITE (NTAPE) ((A(L,K),K=1,8),L=1,NPC)
      LINE=NLINE
      RETURN
      END
```

```
*DECK R
      SUBROUTINE KURFIT (X,Y,A,N,DY1,DY2,K1,K2)
      DIMENSION X(1),Y(1),A(1),B(100)
      N1=N-2
      A(1)=-DY1/2.
      B(1)=X(1)-X(2)
      IF (K1.EQ.2) GO TO 1
      A(1)=-(DY1+(Y(2)-Y(1))/B(1))/B(1)
      B(1)=0.0
1     J=1
      DO 2 I=1,N1
      K=I+1
      J=J+1
      C1=X(K)-X(I)
      C2=X(K+1)-X(K)
      C3=Y(K)-Y(I)
      C4=Y(K+1)-Y(K)
      C5=C3/C1-C4/C2
      C6=C1/C2
      C7=C1*C2
      B(J)=1.0/(C6*(C1-B(J-1)))
      A(J)=(C5/C2-C6*A(J-1))*B(J)
      J=J+1
      B(J)=1.0/((-C1-C2)/C7-C6*B(J-1))
2     A(J)=(-C5/C7-C6*A(J-1))*B(J)
      A(J+1)=(DY2-C4/C2+C2*A(J))/(C2*(B(J)-C2))
      IF (K2.EQ.2) A(J+1)=(DY2/2.+A(J))/(-2.*C2+B(J))
      J=2*(N-1)
3     J=J-1
      IF (J.LE.0) RETURN
      A(J)=A(J)-B(J)*A(J+1)
      GO TO 3
      END
```

```
*DECK S
      SUBROUTINE KURVE (A,X,Y,XP,YP,N,L)
      DIMENSION A(1),X(1),Y(1)
1     DO 3 I=2,N
      IF ((XP-X(1))*(X(2)-X(1))) 4,4,2
2     IF ((XP-X(1))*(XP-X(I))) 4,3,3
3     CONTINUE
4     K=2*I-3
      C1=XP-X(I-1)
      C2=X(I)-XP
      IF (L.EQ.8) C2=0.
      SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))
      YP=Y(I-1)+(SLOPE+A(K)*C2+A(K+1)*C1*C2)*C1
      RETURN
      END
```

```
*DECK T
      SUBROUTINE CONICF(P,Q,R,S,T,SG,X,Y,YP)                            T0010
      ROOT=SG*SQRT((R*X+S)*X+T)                                         T0020
      Y=P*X+Q+ROOT                                                      T0030
      IF((SG.NE.0.).AND.(ROOT.NE.0.)) GO TO 10
      YP=P                                                              T0050
      GO TO 20                                                          T0060
   10 YP=P+(R*X+S/2.0)/ROOT                                             T0070
   20 RETURN                                                            T0080
      END                                                               T0090
```

```
*DECK U
      SUBROUTINE CUBIC(X,Z,ZX,ZXP,A,B,C,F)                              00000660
      DIMENSION ZX(1),ZXP(1)                                            00000670
      DZ1=ZX(8)-ZX(11)                                                  00000680
      DZ2=ZX(9)-ZX(2)                                                   00000690
      DZC=ZX(10)-ZX(11)                                                 00000700
      DZU=ZX(1)-ZX(2)                                                   00000710
      DZ21=ZX(9)-ZX(8)                                                  00000720
      X1=(ZX(12)-ZX(13))*SQRT(1.0-(DZ1/DZC)**2)+ZX(13)
      X2=(ZX(5)-ZX(4))*SQRT(1.0-(DZ2/DZU)**2)+ZX(4)                     00000740
      DX21=X2-X1                                                        00000750
      IF(X1.GT.ZX(13)) GO TO 20
   10 Z1P=0.0                                                           00000770
      X1Y=0.0                                                           00000780
      Z1PY=0.0                                                          00000790
      GO TO 30                                                          00000800
   20 Z1P=-X1/DZ1*(DZC/(ZX(12)-ZX(13)))**2
      X1P=1.0/Z1P
      X1Y=X1*((ZXP(12)-ZXP(13))/(ZX(12)-ZX(13)))+(ZXP(8)+((ZX(8)-ZX(10))
     * *ZXP(11)-DZ1*ZXP(10))/DZC)/Z1P+ZXP(13)
     *  +2.0*((ZXP(12)-ZXP(13))/(ZX(12)-ZX(13))-(ZXP(10)-ZXP(11))/DZC))
      X1PY=X1P*((ZXP(8)-ZXP(11))/DZ1-X1Y/X1
      Z1PY=-X1PY*Z1P/X1P
   30 Z2P=-X2/DZ2*(DZU/(ZX(4)-ZX(5)))**2                               00000860
      X2P=1.0/Z2P
      X2Y=X2*((ZXP(5)-ZXP(4))/(ZX(5)-ZX(4)))+(ZXP(9)+((ZX(9)-ZX(1))    00000870
     * *ZXP(2)-DZ2*ZXP(1))/DZU)/Z2P+ZXP(4)                              00000880
      X2PY=X2P*((ZXP(9)-ZXP(2))/DZ2-X2Y/X2
     * +2.0*((ZXP(4)-ZXP(5))/(ZX(4)-ZX(5))-(ZXP(1)-ZXP(2))/DZU))
      Z2PY=-X2PY*Z2P/X2P
      ZY21=ZXP(9)-ZXP(8)                                                00000930
      DX21Y=X2Y-X1Y                                                     00000940
      IF((Z1P.LT.-4.0).OR.(Z2P.LT.-4.0).OR.(Z1P.GT.0.01)) GO TO 40
      CA=3.0*DZ21-(Z2P+2.0*Z1P)*DX21                                    00000910
      CB=-2.0*DZ21+(Z2P+Z1P)*DX21                                       00000920
      CAY=3.0*ZY21-(Z2PY+2.0*Z1PY)*DX21-(Z2P+2.0*Z1P)*DX21Y             00000950
      CBY=-2.0*ZY21+(Z2PY+Z1PY)*DX21+(Z2P+Z1P)*DX21Y                    00000960
      DX1=X-X1                                                          00000970
      DX2=X-X2                                                          00000980
      T=DX1/DX21                                                        00000990
      A=-Z1P-(2.0*CA+3.0*CB*T)*T/DX21                                   00001000
      B=-ZXP(8)-Z1PY*DX1+Z1P*X1Y-(CAY+CBY*T)*T*T                        00001010
     *  -(A+Z1P)*(X2Y*DX1-X1Y*DX2)/DX21                                 00001020
```

```
      C=1.0                                                             00001030
      F=Z-ZX(8)-Z1P*DX1-(CA+CB*T)*T*T                                   00001040
      RETURN                                                            00001050
   40 CA=3.0*DX21-(X2P+2.0*X1P)*DZ21
      CB=-2.0*DX21+(X2P+X1P)*DZ21
      CAY=3.0*DX21Y-(X2PY+2.0*X1PY)*DZ21-(X2P+2.0*X1P)*ZY21
      CBY=-2.0*DX21Y+(X2PY+X1PY)*DZ21+(X2P+X1P)*ZY21
      DZ1=Z-ZX(8)
      DZ2=Z-ZX(9)
      T=DZ1/DZ21
      A=1.0
      C=-X1P-(2.0*CA+3.0*CB*T)*T/DZ21
      B=-X1Y-X1PY*DZ1+X1P*ZXP(8)-(CAY+CBY*T)*T*T
     * -(C+X1P)*(ZXP(9)*DZ1-ZXP(8)*DZ2)/DZ21
      F=X-X1-X1P*DZ1-(CA+CB*T)*T*T
      RETURN
      END                                                               00001060
```

```
*DECK V
      SUBROUTINE ELIPSE(X,Z,ZX,ZXP,IZK,IZM,IXK,IXM,A,B,C,F)
      DIMENSION ZX(1),ZXP(1)
      FF=-ABS(Z-ZX(IZK))
      DX=X-ZX(IXK)
      DXK=ZX(IXM)-ZX(IXK)
      DZ=Z-ZX(IZM)
      DZK=ZX(IZK)-ZX(IZM)
      A=2.0*DX/DXK**2
      C=2.0*DZ/DZK**2
      B=C*(ZXP(IZM)*FF-ZXP(IZK)*DZ)/DZK+A*(ZXP(IXK)*(X-ZX(IXM))
     *   -ZXP(IXM)*DX)/DXK
      F=(DZ/DZK)**2+(DX/DXK)**2-1.0
      RETURN
      END
```

```
*DECK W
      SUBROUTINE KONIC(X,Z,ZX,ZXP,ICX,ICZ,ISX,ISZ,IB,A,B,C,F)                   00001530
      DIMENSION ZX(1),ZXP(1)                                                    00001540
      SIG = SIGN(1.,ZX(ICZ)-ZX(ISZ))                                            00001550
      H = SIG*(1.414213562*ZX(IB)-ZX(ISX))-ZX(ISZ)                              00001560
      HY = SIG*(1.414213562*ZXP(IB)-ZXP(ISX))-ZXP(ISZ)                          00001570
      ZZ = Z-ZX(ISZ)                                                            00001580
      XX = X-ZX(ISX)                                                            00001590
      Z1 = ZX(ICZ)-ZX(ISZ)                                                      00001600
      Z1Y = ZXP(ICZ)-ZXP(ISZ)                                                   00001610
      X1 = ZX(ICX)-ZX(ISX)                                                      00001620
      X1Y = ZXP(ICX)-ZXP(ISX)                                                   00001630
      CK = (H-Z1)**2/(4.*H*X1*Z1*(H-SIG*X1-Z1))                                 00001640
      CKY = CK*(2.*(HY-Z1Y)/(H-Z1)-HY/H-X1Y/X1-Z1Y/Z1-                          00001650
     1             (HY-SIG*X1Y-Z1Y)/(H-SIG*X1-Z1))                              00001660
      XZX = X1*ZZ-Z1*XX                                                         00001670
      F = CK*XZX*XZX+XX*(ZZ-Z1)                                                 00001680
      A = -2.*CK*Z1*XZX+ZZ-Z1                                                   00001690
      C = 2.*CK*X1*XZX+XX                                                       00001700
      B = -A*ZXP(ISX)-C*ZXP(ISZ)+XZX*XZX*CKY+2.*CK*ZZ*XZX*X1Y                   00001710
     1    -(2.*CK*XZX+1.)*XX*Z1Y                                                00001720
      RETURN                                                                    00001750
      END                                                                       00001760
```

```
*DECK X
      SUBROUTINE AERO(NPC,SINA,COSA)
      COMMON /CP/B(50,4,2),DRAG,ALIFT,TORQUE,YMT,ZMT,WINGA
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3       A   6
      DO 10 I=1,2
      DO 10 J=2,NPC
      DO 10 K=1,2
      L=J+K-1
      M=(3-K)*I+(3-I)*K-3
      CPA=(B(L,4,M)+B(J+1,4,I)+B(J,4,3-I))/(6.0*WINGA)
      YA =(B(L,2,M)+B(J+1,2,I)+B(J,2,3-I))/3.0
      ZA =(B(L,3,M)+B(J+1,3,I)+B(J,3,3-I))/3.0
      IF(I.NE.K) CPA=-CPA
      DARY=CPA*((B(J+1,3,I)-B(L,3,M))*(B(J,1,3-I)-B(L,1,M))-
     *           (B(J+1,1,I)-B(L,1,M))*(B(J,3,3-I)-B(L,3,M)))
      DARZ=CPA*((B(J+1,1,I)-B(L,1,M))*(B(J,2,3-I)-B(L,2,M))-
     *              (B(J+1,2,I)-B(L,2,M))*(B(J,1,3-I)-B(L,1,M)))
      DRAG=DRAG+DARY*COSA+DARZ*SINA
      ALIFT=ALIFT+DARZ*COSA-DARY*SINA
   10 TORQUE=TORQUE+DARY*(ZA-ZMT)-DARZ*(YA-YMT)
      FY=DRAG*COSA-ALIFT*SINA
      FZ=DRAG*SINA+ALIFT*COSA
      WRITE(IOU6,600) DRAG,ALIFT,TORQUE,YMT,ZMT,FY,FZ
  600 FORMAT(1H0,T7,4HDRAG,T22,4HLIFT,T35,6HMOMENT,T47,
     $ 13HPIVOT POINT Y,T68,1HZ,T80,7HY-FORCE,T95,7HZ-FORCE/7G15.6)
      RETURN
      END
```

```
*DECK Y
      SUBROUTINE SETCP (LNTAPE,IB,A,B,SINE,COSINE,DUMMY,GF,FMSTR,N2)
      DIMENSION A(50,8),B(50,4,2)
      REWIND LNTAPE
      READ (LNTAPE) ((A(L,M),M=1,8),L=2,N2)
      DO 10 I=2,N2
      B(I,1,IB) = COSINE*A(I,1)-SINE*A(I,2)
      B(I,2,IB) = SINE*A(I,1)+COSINE*A(I,2)
      B(I,3,IB) = A(I,3)
   10 B(I,4,IB) = 2.0*(A(I,7)*DUMMY-1.0)/(GF*FMSTR*FMSTR)
      REWIND LNTAPE
      RETURN
      END
```

```
      PROGRAM FUSLAG(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PUNCH,TAPE7=
     1PUNCH,TAPE2,TAPE1,TAPE4,TAPE3)
      COMMON /MESH/ K1(8),AL,Z2L,ZX23,ZX7,SIDEP,X2L
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST,KPRINT(3)
      COMMON /GEOM/ YMAX,YSTOP,UNIL,INDEX,DBLNL,NCQ,CQ(8,10)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE,ISURF,YO,ISTART,WINGST
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      COMMON /STOR/ H(4,8),A(99,8,4)
      COMMON /CP/ B(99,4,2),DRAG,ALIFT,TORQUE,YMT,ZMT,WINGA
      DIMENSION YSET(20),NPSET(180),LIT(18),DUM(10),L1(8)
C     IN A(I,J,K) J=1,8 CORRESPOND TO X,Y,Z,U,V,W,P,PT, K=1,4 CORRESPOND
C     TO A(ACQUIRER), L(LOWER CLASS), M(MIDDLE CLASS), U(UPPER CLASS)
      CALL FTNBIN (1,0,DUM)
      IOU5=5
      INCASE=3
      IOU6=6
      IOU7=7
      IOU3=1
      IOU=IOU5
      LTAPE=4
      NTAPE=2
      REWIND LTAPE
      REWIND NTAPE
      LN=LTAPE+NTAPE
      READ (IOU5,640) (LIT(I),I=1,18)
      READ (IOU5,670) DP,DTHETA,DPSI,EPSLN,DYC,YSTOP,STEPM,WINGST
      READ (IOU5,680) NSURF,LINE,NPC,NR,IFT,ISTART,KP4,NS,NL,NP,KPRINT
      IOU1=IOU3
      IF (KP4.LT.0) IOU1=IOU6
      KAB=IABS(KP4)
      IF (KP4.EQ.0) KAB=10
      DO 10 I=1,NR
   10 READ (IOU5,650) YSET(I),NPSET(9*I-8),NPSET(9*I-7),NPSET(9*I-6),
     1NPSET(9*I-5),NPSET(9*I-4),NPSET(9*I-3),NPSET(9*I-2),NPSET(9*I-1),
     2NPSET(9*I)
```

```
      READ (IOU5,670) FMSTR,GF(1),ALFA,YO
      READ (IOU5,670) YA,ZA,SCALE,WINGA,DBLNL
      READ (IOU5,680) NCQ
      DO 20 J=1,NCQ
      READ (IOU5,810) IC,(CQ(I,J),I=1,2)
      READ (IOU5,810) IC,(CQ(I,J),I=3,8)
20    CONTINUE
      WRITE (IOU6,660) (LIT(I),I=1,18)
      WRITE (IOU6,670) DP,DTHETA,UPSI,EPSLN,DYC,YSTOP,STEPM,WINGST
      WRITE (IOU6,680) NSURF,LINE,NPC,NR,IFT,ISTART,KP4,NS,NL,NP,KPRINT
      DO 30 I=1,NR
30    WRITE (IOU6,650) YSET(I),NPSET(9*I-8),NPSET(9*I-7),NPSET(9*I-6),
     1NPSET(9*I-5),NPSET(9*I-4),NPSET(9*I-3),NPSET(9*I-2),NPSET(9*I-1),
     2NPSET(9*I)
      WRITE (IOU6,670) FMSTR,GF(1),ALFA,YO
      WRITE (IOU6,670) YA,ZA,SCALE,WINGA,DBLNL
      WRITE (IOU6,680) NCQ
      JJ=22
      DO 40 J=1,NCQ
      IC=CQ(8,J)
      WRITE (IOU6,820) J,(CQ(I,J),I=1,2)
      WRITE (IOU6,820) JJ,(CQ(I,J),I=3,7),IC
40    CONTINUE
      WINGA=WINGA*144.0
      STEP=0.
      MESS=1
      INCRES=0
      GF(2)=.5*(GF(1)-1.)
      GF(3)=.5*(GF(1)+1.)
      GF(4)=GF(2)/GF(3)
      GF(5)=1./(GF(1)-1.)
      GF(6)=GF(1)*GF(5)
      GF(7)=1./GF(6)
      GFM2=GF(2)*FMSTR*FMSTR
      QINF=SQRT(GFM2/(1.+GFM2))
      POP=(1.+GFM2)**GF(6)
      RAD=57.295779513082
      COSA=COS(ALFA/RAD)
      SINA=SIN(ALFA/RAD)
      ISURF=1
```

```
      ILL=0
      KURE=0
      DO 50 I=1,8
50    K1(I)=0
      SINE=0.
      COSINE=1.
      S2=0.
      C2=1.
      SN=SIN(.1)
      CS=COS(.1)
      YMAX=-1.E7
C ISTART.LT.0,RESTART FROM CARDS,ISTART=I.GT.0,RESTART FROM ITH SURFACE
      IF (ISTART.NE.0.OR.IFT.NE.1) GO TO 60
      IOU=LTAPE
      REWIND IOU3
      READ (IOU,670) FMSTR,GF(1),ALFA,YI,DRAG$YO=(YI-1.0)*SCALE+YA
      DRAG=DRAG*3.1415927*SCALE**2/WINGA
60    IF (ISTART.EQ.0) REWIND IOU3
      IF (ISTART.NE.0.OR.KP4.LE.0) GO TO 70
      WRITE (IOU3,800) (LIT(I),I=1,18),FMSTR,GF(1),ALFA
      WRITE (IOU3,740) ISURF,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      WRITE (IOU3,780) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE
70    IF (ISTART) 120,130,80
80    DO 100 I=1,ISTART
      READ (IOU3,640) DUM
      IF (KPRINT(3).EQ.3) WRITE (IOU6,640) DUM
      READ (IOU3,770) ISURF,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      IF (KPRINT(3).NE.0) WRITE (IOU6,680) ISURF,LINE,NPC,K1
      READ (IOU3,780) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE
      READ (IOU3,640) DUM
      IF (IFT.LT.0) INCRES=-IFT
      IF (IFT.GT.1) INCRES=1-IFT
      IF (ISURF.EQ.ISTART) GO TO 130
      KOUNT=LINE*(NPC+2)+100
      DO 90 J=1,KOUNT
      READ (IOU3,640) DUM
      IF (KPRINT(3).EQ.3) WRITE (IOU6,640) DUM
      IF (EOF(IOU3)) 100,90
90    CONTINUE
100   CONTINUE
```

```
110   WRITE (IOU6,760) ISURF,ISTART
      CALL EXIT
120   READ (IOU5,680) ISURF,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      READ (IOU5,690) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE
130   N1=ISURF+1
      N2=NPC+1
      N3=NPC+3
      WRITE (IOU6,700) FMSTR,GF(1),ALFA,DRAG,ALIFT,TORQUE
      DO 170 I=1,LINE
      IF (ISTART.GT.0) READ (IOU3,640) DUM
      IND=2
      IF (I.EQ.1) IND=1
      IF (I.EQ.LINE) IND=3
      IF (ISTART.GT.0) READ (IOU3,640) DUM
      DO 160 J=2,N2
      IF (ISTART.LE.0) READ (IOU,710) A(J,3),A(J,1),A(J,7),A(J,8),THETA,
     1PSI
      IF (ISTART.GT.0) READ (IOU3,790) A(J,3),A(J,1),A(J,7),A(J,8),THETA
     1,PSI
      THETA=THETA/RAD
      PSI=PSI/RAD
      A(J,7)=A(J,8)/(1.+GF(2)*A(J,7)**2)**GF(6)
      IF (ISTART.NE.0) GO TO 140
      A(J,1)=A(J,1)*SCALE
      A(J,3)=A(J,3)*SCALE+ZA
140   A(J,2)=YO
      Q=SQRT(1.-(A(J,7)/A(J,8))**GF(7))
      QS=Q*SIN(THETA)
      A(J,4)=QS*COS(PSI)
      A(J,5)=Q*COS(THETA)
      A(J,6)=QS*SIN(PSI)
      DO 150 K=1,4,3
150   CALL ROTATE (A(J,K),A(J,K+1),SINE,COSINE)
      CALL DATOUT (A,J,IND,KAB,0)
160   CALL DATOUT (A,J,IND,KAB,1)
170   WRITE (NTAPE) ((A(J,L),L=1,8),J=2,N2)
      IF (ISTART.EQ.0.AND.KP4.GT.0) END FILE IOU3
      IF (ISTART.LE.0) GO TO 180
      READ (IOU3,640) DUM
      IF (EOF(IOU3)) 180,110
```

```
  180 DO 630 I=N1,NSURF                                                   A 161
      ITEST=0                                                             A 162
      KTEST=0                                                             A 163
      ISURF=I                                                             A 164
      NLINE=LINE+INCRES                                                   A 165
  190 DYC=(DYC+DYC*STEP)*FLOAT(LINE)/FLOAT(NLINE)                         A 166
      IF(YO.LE.WINGST-.5.AND.(WINGST.LE.YO+DYC.AND.YO+DYC.LE.WINGST
     1      +1.)) DYC=WINGST-.01-YO
      IF (STEP.LT.0.DO) MESS=1                                            A 167
      STEP=.05                                                            A 168
      IPRINT=0                                                            A 169
      IF (I.EQ.NS) IPRINT=KPRINT(3)                                       A 170
      DO 210 K=1,NR                                                       A 171
      NPCC=NPSET(9*(K-1)+1)                                               A 172
      DO 200 J=1,8                                                        A 173
  200 L1(J)=NPSET(9*(K-1)+J+1)                                            A 174
      IF (YO+DYC.LT.YSET(K).OR.NPSET(9*K+1).EQ.0) GO TO 220               A 175
  210 CONTINUE                                                            A 176
  220 IF (NPC.EQ.NPCC.AND.(L1(1).EQ.K1(1).AND.L1(2).EQ.K1(2))) 230,250    A 177
  230 IF (L1(3).EQ.K1(3).AND.(L1(4).EQ.K1(4).AND.L1(5).EQ.K1(5))) 240,    A 178
     1250                                                                 A 179
  240 IF (L1(6).EQ.K1(6).AND.(L1(7).EQ.K1(7).AND.L1(8).EQ.K1(8))) 290,    A 180
     1250                                                                 A 181
  250 DYC=DYC*FLOAT(2*NPC-2+K1(1)-K1(2)+K1(3)-K1(4)+K1(5)-K1(6)+K1(7)-K1  A 182
     1(8))/FLOAT(2*NPCC-2+L1(1)-L1(2)+L1(3)-L1(4)+L1(5)-L1(6)+L1(7)-L1(8  A 183
     2))                                                                  A 184
      IF (K1(5).EQ.0.AND.L1(5).NE.0) DYC=DYC/FLOAT(K8+1)                  A 185
      IF (YO+DYC.GE.YSET(K-1)) GO TO 270                                  A 186
      NPCC=NPSET(9*(K-2)+J+1)                                             A 187
      DO 260 J=1,8                                                        A 188
  260 L1(J)=NPSET(9*(K-2)+J+1)                                            A 189
  270 DO 280 K=1,8                                                        A 190
  280 K1(K)=L1(K)                                                         A 191
      IF(YO.GT.WINGST-0.5.AND.(WINGST.LE.YO+DYC.AND.YO+DYC.LE.WINGST
     1      +1.)) DYC=WINGST+1.-YO
      MESS=1                                                              A 192
  290 REWIND LTAPE                                                        A 193
      REWIND NTAPE                                                        A 194
      INDEX=-1                                                            A 195
      CALL FIGURE (0.,YO,0.,F,A1,B1,C1)                                   A 196
```

```
      INDEX=-2
      IF ((YO+DYC.LE.YSTOP).OR.(ABS(YSTOP-YO).LT.0.0002)) GO TO 300
      DYC=YSTOP-YO
      KP4=-10000
300   CALL FIGURE (0.,YO+DYC,0.,F,A1,B1,C1)
      IF (ITEST.EQ.KTEST) GO TO 310
      IF (ITEST.GT.10) CALL EXIT
      KTEST=ITEST
      GO TO 380
310   LTAPE=LN-LTAPE
      NTAPE=LN-NTAPE
      IF (MESS.EQ.0.AND.INCRES.EQ.0) GO TO 330
      DO 320 K=1,LINE
      READ (LTAPE) ((A(L,M),M=1,8),L=1,NPC)
      IF (K.EQ.1) AM=.5*(A(1,3)+A(NPC,3))
      CALL TIDYUP (SINE,COSINE,A,NPC,NPCC,AM,K)
320   WRITE (NTAPE) ((A(L,M,2),M=1,8),L=1,NPCC)
      WRITE (IOU6,750) I
      NPC=NPCC
      N2=NPC+1
      N3=NPC+3
      CALL REFORM (A,LTAPE,NTAPE,NPC,LINE,NLINE)
      INCRES=0
      MESS=0
      GO TO 290
330   IF (KURE.NE.ILL) 350,380
340   REWIND LTAPE
      REWIND NTAPE
350   WRITE (IOU6,730) ILL,I
      IF (ILL.GT.16) CALL EXIT
      KURE=ILL
      CALL ROTATE (SINE,COSINE,SN,CS)
      S2=2.*SINE*COSINE
      C2=COSINE*COSINE-SINE*SINE
      DO 370 J=1,LINE
      READ (LTAPE) ((A(L,K),K=1,8),L=2,N2)
      DO 360 K=2,N2
      DO 360 L=1,4,3
360   CALL ROTATE (A(K,L),A(K,L+1),SN,CS)
370   WRITE (NTAPE) ((A(L,K),K=1,8),L=2,N2)
```

```
      REWIND LTAPE
      REWIND NTAPE
      LTAPE=LN-LTAPE
      NTAPE=LN-NTAPE
380   IF (KP4.LE.1.OR.I/KP4*KP4.EQ.I) WRITE (IOU6,720) I
      CALL SETCP (LTAPE,1,A,B,SINE,COSINE,POP,GF,FMSTR,N2)
      IF (STEPM.NE.0.) DYC=AMIN1(DYC,STEPM)
      DO 570 J=1,LINE
      REWIND INCASE
      WRITE (INCASE,830)
      IND=2
      IF (J.EQ.1) IND=1
      IF (J.EQ.LINE) IND=3
      IF (IND-2) 390,430,460
390   DO 400 M=3,4
400   READ (LTAPE) ((A(K,L,M),L=1,8),K=2,N2)
      IF (DYC.GT.1.E-6) GO TO 410
      DYC=AMIN1(ABS(A(2,3,3)-A(2,3,4)),ABS(A(3,1,3)-A(2,1,3)))
      XM2=(1.*(A(2,8,3)/A(2,7,3))**GF(7)-1.)/GF(2)
      DYC=DYC*.6*SQRT(XM2-1.)
410   CALL MIRROR (A(1,1,3),S2,C2,N2)
      DO 420 K=2,NPC
      IF (.6*((A(K,1,4)-A(K,1,3))**2+(A(K,2,4)-A(K,2,3))**2+(A(K,3,4)-A(
     1K,3,3))**2).LT.(A(K,1,3)-A(K+1,1,3))**2+(A(K,2,3)-A(K+1,2,3))**2+(
     2A(K,3,3)-A(K+1,3,3))**2) GO TO 450
420   CONTINUE
      INCRES=1
      GO TO 450
430   DO 440 K=1,N3
      DO 440 L=1,8
      A(K,L,2)=A(K,L,3)
440   A(K,L,3)=A(K,L,4)
      READ (LTAPE) ((A(L,K,4),K=1,8),L=2,N2)
450   CALL MIRROR (A(1,1,4),S2,C2,N2)
      GO TO 480
460   DO 470 K=3,4
      DO 470 L=1,N3
      DO 470 M=1,4,3
      YTEMP=SINE*A(L,M,K)+COSINE*A(L,M+1,K)
      A(L,M,K-1)=COSINE*A(L,M,K)-SINE*A(L,M+1,K)
```

```
      A(L,M+1,K-1)=COSA*YTEMP+SINA*A(L,M+2,K)
      A(L,M+2,K-1)=-SINA*YTEMP+COSA*A(L,M+2,K)
470   A(L,M/4+7,K-1)=A(L,M/4+7,K)
480   DO 500 K=2,N2
      IPRINT=0
      IF (I.NE.NS) GO TO 490
      IPRINT=KPRINT(1)
      IF (J.NE.NL) GO TO 490
      IPRINT=KPRINT(2)
      IF (K.NE.MP+1) GO TO 490
      IPRINT=KPRINT(3)
490   CALL BULK (K,IND)
      IF (ITEST.NE.KTEST) GO TO 190
      IF (KURE.NE.ILL) GO TO 340
      IF (IND.NE.3) CALL DATOUT (A,K,IND,KAB,0)
      KAY=K-1
      WRITE (INCASE,840) KAY,J
500   CONTINUE
      IF (IND.NE.3) WRITE (NTAPE) ((A(L,K),K=1,8),L=2,N2)
      IF (IND-2) 540,570,510
510   CALL HARNES (QINF,A,NPC)
      REWIND INCASE
      END FILE INCASE
      DO 530 K=2,4
      N=K
      IF (K.EQ.4) N=1
      DO 530 L=1,N3
      IF (K.EQ.4.AND.(L.EQ.1.OR.L.EQ.N3)) GO TO 530
      DO 520 M=1,4,3
      A(L,M/4+7,K)=A(L,M/4+7,N)
      YTEMP=COSA*A(L,M+1,N)-SINA*A(L,M+2,N)
      A(L,M+2,K)=SINA*A(L,M+1,N)+COSA*A(L,M+2,N)
      A(L,M+1,K)=-SINE*A(L,M,N)+COSINE*YTEMP
520   A(L,M,K)=COSINE*A(L,M,N)+SINE*YTEMP
530   CONTINUE
      GO TO 570
540   LB=2
      IF (K1(1).NE.0) LB=K1(1)+1
      LE=NPC
      IF (K1(2).NE.0) LE=K1(2)
```

```
      DO 560 L=LB,LE
      ARC=(A(L+1,1)-A(L,1))**2+(A(L+1,2)-A(L,2))**2+(A(L+1,3)-A(L,3))**2
      IF (L.NE.LB) GO TO 550
      BIG=ARC
      SMALL=ARC
550   BIG=AMAX1(BIG,ARC)
560   SMALL=AMIN1(SMALL,ARC)
      IF (2.*SMALL.LE.BIG) MESS=1
570   CONTINUE
      DO 590 J=2,N2
      DO 580 K=1,8
580   A(J,K)=A(J,K,4)
      CALL DATOUT (A,J,3,KAB,0)
590   CONTINUE
      IF (KP4.GT.0.OR.(KP4.EQ.0.AND.I/10*10.EQ.I)) WRITE (IOU3,720) I
      WRITE (NTAPE) ((A(J,L),L=1,8),J=2,N2)
      CALL SETCP (NTAPE,2,A,B,SINE,COSINE,POP,GF,FMSTR,N2)
      CALL AERO (NPC,SINA,COSA)
      REWIND NTAPE
      READ (NTAPE) A(1),A(2)
      YO=SINE*A(1)+COSINE*A(2)
      IF (KP4.EQ.0.AND.I/10*10.NE.I) GO TO 600
      WRITE (IOU1,740) I,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      WRITE (IOU1,780) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,ALIFT,TORQUE
      IF (I/KAB*KAB.NE.I.AND.KP4.NE.0) GO TO 610
600   IF (KP4.GE.0) WRITE (IOU6,740) I,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      IF (KP4.LT.0) WRITE (IOU7,680) I,LINE,ILL,KURE,MESS,INCRES,NPC,K1
      IF (KP4.GE.0) WRITE (IOU6,780) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,
     1ALIFT,TORQUE
      IF (KP4.LT.0) WRITE (IOU7,690) SINE,COSINE,S2,C2,DYC,STEP,YO,DRAG,
     1ALIFT,TORQUE
610   IF (KP4.LT.1.AND.I/KAB*KAB.NE.I) GO TO 630
      REWIND NTAPE
      DO 620 J=1,LINE
      READ (NTAPE) ((A(K,L),L=1,8),K=2,N2)
      IND=2
      IF (J.EQ.1) IND=1
      IF (J.EQ.LINE) IND=3
      DO 620 K=2,N2
620   CALL DATOUT (A,K,IND,KAB,1)
```

```
      IF (KP4.GT.0.OR.(KP4.EQ.0.AND.I/10*10.EQ.I)) END FILE IOU3
  630 CONTINUE
      CALL EXIT
C
C
C
  640 FORMAT (18A4)
  650 FORMAT (F10.5,12I5)
  660 FORMAT (1H140X38HNORTHROP SUPERSONIC FLOW FIELD PROGRAM//30X18A4)
  670 FORMAT (8F10.5)
  680 FORMAT (16I5)
  690 FORMAT (4E15.8)
  700 FORMAT (1H039X40H 3-DIMENSIONAL CHARACTERISTICS SOLUTION /1H05X,24
     1HFREESTREAM MACH NUMBER =,F7.3,5X25HRATIO OF SPECIFIC HEATS =,F6.3
     2,5X17HANGLE OF ATTACK =,F7.3,8H DEGREES/1H049X21HINITIAL VALUE SUR
     3FACE,T2,6HDRAG =,G14.7/T2,6HLIFT =,G14.7/T2,8HMOMENT =,G14.7)
  710 FORMAT (2E15.8,F10.7,E15.8,F12.8,F13.8)
  720 FORMAT (1H150X15HDATA SURFACE NO,I3/93H0***TO USE THIS AS AN INITI
     1AL VALUE SURFACE, KEY PUNCH FIRST 6 COLUMNS OF DATA AS INPUT CARDS
     2)
  730 FORMAT (33H0ILLNATURED PSI AND DELTA,CASE NO,I3,42H, ROTATE COORDI
     1NATES AND RECOMPUTE SURFACE,I3)
  740 FORMAT (61H0TO USE AS IVS, PUNCH THE 4 CARDS, PLACE BEFORE BODY PT
     1 CARDS/1X,9(1H0),10(1H1),10(1H2),10(1H3),10(1H4),10(1H5),10(1H6),
     210(1H7),1H8/1X,8(10H1234567890)//1X,20I5)
  750 FORMAT (42H0DATA POINTS WILL BE RE-SPACED FOR SURFACE,I3)
  760 FORMAT (13H0WRONG IVS NOI4,4H, NOI4,27H SPECIFIED OR FAIL END FILE
     1)
  770 FORMAT (1X/////1X,20I5)
  780 FORMAT (1X,4E15.8)
  790 FORMAT (1X,2E15.8,F10.7,E15.8,F12.8,F13.8)
  800 FORMAT (1H1,30X18A4,/1H05X24HFREESTREAM MACH NUMBER =F7.3,5X25HRAT
     1IO OF SPECIFIC HEATS =F6.3,5X,17HANGLE OF ATTACK =F7.3,8H DEGREES)
  810 FORMAT (I2,5E15.8,F2.0)
  820 FORMAT (1X,I2,1P5E15.8,I2)
  830 FORMAT (1H1,40X,41H* * * * * * * * * * * * * * * * * * * * */41X,
     141H*                                   */41X,41H*      IN THE
     2EVENT OF FAILURE            */41X,41H*
     3          */41X,41H*   I WAS DOING O. K.                  */41X,41
     4H*   NOT SO--THE NEXT GUY.                */41X,41H*      HE GOOFS I
```

```
5N EVERY POSSIBLE WAY.   */41X,41H*   I WAS DOING O. K.                     A 397
6       */41X,41H*     BUT THINGS WENT ASTRAY.              */41X,41H*   A 398
7   NOW THE LISTING CAN TESTIFY:         */41X,41H*   I WAS DOING O.      A 399
8 K.                       */41X,41H*   NOT SO--THE NEXT GUY.              A 400
9      */41X,41H*                                           */41X,41H* *   A 401
A * * * * * * * * * * * * * * * * * * * * *///)                            A 402
840   FORMAT (41X,17H    I AM POINT NO,I3,11H OF LINE NO,I3)               A 403
      END                                                                  A 404
```

```
      SUBROUTINE BULK (L,IND)
C     IND =1 FOR BODY, 2 FOR FIELD, 3 FOR SHOCK
C**** TAKE NOTE THAT SINE,COSINE MUST CORRESPOND TO CURRENT  COORDINATES
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      COMMON /STOR/ H(4,8),XA1(99),YA1(99),ZA1(99),UA1(99),VA1(99),WA1(
     199),PA1(99),PPMA1(99)
      DIMENSION XYZ(3,7)
      EQUIVALENCE (XC,XYZ)
10    IF (IPRINT.GE.1) 20,30
20    IF (IND.EQ.1) WRITE (IOU6,150)
      IF (IND.EQ.2) WRITE (IOU6,140)
      IF (IND.EQ.3) WRITE (IOU6,160)
      N=4-MOD(IND,2)
30    CALL PICK (L,IND)
      IF (KURE.NE.ILL) RETURN
C     BACK TO MAIN PROGRAM, ROTATE Y AXIS AWAY FROM FLOW FIELD
      UC=HUB(4)
      VC=HUB(5)
      WC=HUB(6)
      PC=HUB(7)
      PPMC=HUB(8)
      THETAC=TH
      PSIC=PS
      DO 40 I=1,4
40    LOT(I)=0
      DO 120 I=1,25
      IF (IPRINT.GE.1) WRITE (IOU6,170) I
      IF (IND.NE.3) GO TO 50
      Q=SQRT(1.-(PC/PPMC)**GF(7))
      CALL NORM (-UC*Q,QINF-VC*Q,-WC*Q,A,B,C)
      IF (I.EQ.1) CALL NORM (A,B,C,A1,B1,C1)
      CALL NORM (A+A1,B+B1,C+C1,A,B,C)
      IF (IPRINT.GE.1) WRITE (IOU6,220) A,B,C,A1,B1,C1
```

```
      A4=-A*B
      B4=1.-B*B
      C4=-C*B
      BM=COSA*B4-SINA*C4
      AB=(SINA*B4+COSA*C4)/BM
      XC=HUB(1)+DYC*A4/BM
      YC=HUB(2)+DYC*(COSA+SINA*AB)
      ZC=HUB(3)+DYC*(-SINA+COSA*AB)
      GO TO 60
50    A4=UC+HUB(4)
      B4=VC+HUB(5)
      C4=WC+HUB(6)
      BM=SINE*A4+COSINE*B4
      AB=(COSINE*A4-SINE*B4)/BM
      XC=HUB(1)+DYC*(SINE+COSINE*AB)
      YC=HUB(2)+DYC*(COSINE-SINE*AB)
      ZC=HUB(3)+DYC*C4/BM
60    IF (IPRINT.GE.1) WRITE (IOU6,220) A4,B4,C4,BM,AB,XC,YC,ZC
      IF (IND.NE.1) GO TO 70
      CALL GMTRY (SINE,COSINE,XC,YC,ZC,A4,B4,C4)
      IF (IPRINT.GE.1) WRITE (IOU6,220) XC,YC,ZC,A4,B4,C4
      CALL TRANS (H,XYZ,A4,B4,C4,1.,1)
70    CALL BASEPT (IND)
      IF (ITEST.EQ.KTEST.OR.LOT(4).NE.5) GO TO 80
      H(4,1)=XA1(L)
      H(4,2)=YA1(L)
      H(4,3)=ZA1(L)
      CALL NORM (UA1(L),VA1(L),WA1(L),H(4,4),H(4,5),H(4,6))
      H(4,7)=PA1(L)
      H(4,8)=PPMA1(L)
      KTEST=ITEST
      GO TO 70
80    IF (KURE.NE.ILL.OR.ITEST.NE.KTEST) RETURN
      IF (IPRINT.GE.1) WRITE (IOU6,180) (J,XBS(J),YBS(J),ZBS(J),PBS(J),
     1PPMBS(J),THBS(J),PSBS(J),UBS(J),VBS(J),WBS(J),J=1,N)
      CALL COEFS (IND)
      IF (KURE.NE.ILL) RETURN
      IF (IND.EQ.1) CALL SOL (COF(4),COF(6),COF(5),COF(1),COF(6),COF(9),
     1COF(8),COF(3),0.,0.,1.,1.5707963267949,PC,PSIC,THETAC)
      IF (IND.EQ.2) CALL SOL (COF(4),COF(5),COF(6),COF(1),COF(5),COF(7),
```

```
     1COF(8),COF(2),COF(6),COF(8),COF(9),COF(3),PC,THETAC,PSIC)
      IF (IND.EQ.3) CALL COMPAT
      SINTH=SIN(THETAC)
      UC=SINTH*COS(PSIC)
      VC=COS(THETAC)
      WC=SINTH*SIN(PSIC)
      IF (IND.EQ.1) CALL TRANS (H,XYZ,A4,B4,C4,-1.,1)
      IF (UC*UC+WC*WC.GE.4.E-06) GO TO 90
      ILL=ILL+1
      RETURN
90    IF (IPRINT.GE.1) WRITE (IOU6,190) I,XC,YC,ZC,PC,PPMC,UC,VC,WC,
     1THETAC,PSIC
      IF (L.NE.2.AND.L.NE.NPC+1) GO TO 100
      UT=SINE*UC+COSINE*VC
      IF (IND.NE.3) CALL NORM (SINE*UT,COSINE*UT,WC,UC,VC,WC)
      IF (IND.EQ.3) CALL NORM (0.,VC,WC,UC,VC,WC)
100   IF (I.EQ.1) GO TO 110
      IF (ABS(THETAC-THTEMP).GT.DTHETA) GO TO 110
      IF (ABS(PSIC-PSTEMP).GT.DPSI) GO TO 110
      IF (ABS((PC-PTEMP)/PC).LE.DP) GO TO 130
110   THTEMP=THETAC
      PSTEMP=PSIC
120   PTEMP=PC
      WRITE (IOU6,200)
      CALL EXIT
130   L1=L-1
      IF (IPRINT.GE.1) WRITE (IOU6,210) L1
      Q=SQRT(1.-(PC/PPMC)**GF(7))
      XA1(L)=XC
      YA1(L)=YC
      ZA1(L)=ZC
      UA1(L)=Q*UC
      VA1(L)=Q*VC
      WA1(L)=Q*WC
      PA1(L)=PC
      PPMA1(L)=PPMC
      RETURN
C
C
C
```

```
140   FORMAT (18HO***** FIELD *****)
150   FORMAT (17HO***** BODY *****)
160   FORMAT (18HO***** SHOCK *****)
170   FORMAT (1HO,20X,32HBASE POINT PROPERTIES, ITERATION,I3)
180   FORMAT (15HOI,X,Y,Z,P,PPM=,I5,5E18.8/17H THETA,PSI,U,V,W=,3X5E18.8
     1)
190   FORMAT (41HONEW BULK POINT PROPERTIES, ITERATION NO.,I3/1H 21X,2HX
     1C,18X,2HYC,18X,2HZC,18X,2HPC,17X,4HPPMC/1H ,10X,5E20.8/1H ,21X,2HU
     2C,18X,2HVC,18X,2HWC,16X,6HTHETAC,15X,4HPSIC/1H ,10X,5E20.8)
200   FORMAT (54HOBULK POINT HAS FAILED TO CONVERGE AFTER 25 ITERATIONS)
210   FORMAT (24HO******** BULK POINT NO.,I3,23H HAS CONVERGED ********)
220   FORMAT (8E15.7)
      END
```

```
      SUBROUTINE PICK (L,IND)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /STOR/ H(4,8),ALMUS(99,8,4)
10    LP=L+1
      LM=L-1
      N=2
      IF (IND.EQ.2) N=4
      DO 20 I=1,8
      HUB(I)=ALMUS(L,I,3)
      IF (IND.NE.3) H(2,I)=ALMUS(L,I,4)
      IF (IND.NE.1) H(N,I)=ALMUS(L,I,2)
      H(1,I)=ALMUS(LM,I,3)
20    H(3,I)=ALMUS(LP,I,3)
      CALL NORM (HUB(4),HUB(5),HUB(6),HUB(4),HUB(5),HUB(6))
      IF (HUB(4)**2+HUB(6)**2.GT.4.E-06) GO TO 30
      ILL=ILL+1
      RETURN
30    TH=ACOS(HUB(5))
      PS=ATAN2(HUB(6),HUB(4))
      IF (IND.NE.2) N=3
      DO 40 I=1,N
40    CALL NORM (H(I,4),H(I,5),H(I,6),H(I,4),H(I,5),H(I,6))
      IF (IPRINT.GE.2) WRITE (IOU6,50) HUB,((H(I,J),J=1,8),I=1,N),TH,PS
      RETURN
C
C
C
50    FORMAT (17H0***** PICK *****/(8E15.7))
      END
```

```
      SUBROUTINE BASEPT (IND)
      COMMON /TEST/ LOT(4),ITEST,STEP,KTEST,KPRINT(3)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE,ISURF,YO,ISTART,WINGST
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      COMMON /STOR/ XH(4),YH(4),ZH(4),UH(4),VH(4),WH(4),PH(4),PPMH(4)
      IF (IPRINT.GE.2) WRITE (IOU6,160)
      XD=XC-HUB(1)
      YD=YC-HUB(2)
      ZD=ZC-HUB(3)
      B2=XD*XD+YD*YD+ZD*ZD
      ROOT=GF(2)/((PPMC/PC)**GF(7)-1.0)
      IF (ROOT.GT.0.) GO TO 10
      WRITE (IOU6,120) PPMC,PC
      CALL EXIT
   10 AMU4=ASIN(SQRT(ROOT))
      IF (IPRINT.GE.2) WRITE (IOU6,170) XD,YD,ZD,AMU4
      LID=3
      IF (IND.EQ.2) LID=4
      N=1
      IF (LOT(4).EQ.5) N=4
      DO 100 I=N,LID
      XD1=XH(I)-HUB(1)
      YD1=YH(I)-HUB(2)
      ZD1=ZH(I)-HUB(3)
      IF (IPRINT.GE.2) WRITE (IOU6,180) I
      A2=XD1*XD1+YD1*YD1+ZD1*ZD1
      AB=XD*XD1+YD*YD1+ZD*ZD1
      IF (IPRINT.GE.2) WRITE (IOU6,170) XD1,YD1,ZD1
      DO 70 J=1,50
      IF (J.EQ.1) FL=0.
      IF (J.EQ.1) GO TO 60
      COS2=COS(.5*(AMU0+AMU4))**2
      CALL NORM (.5*(UBS(I)+UC),.5*(VBS(I)+VC),.5*(WBS(I)+WC),U,V,W)
      BV=XD*U+YD*V+ZD*W
      AV=XD1*U+YD1*V+ZD1*W
```

```
      FL=0.
      A=A2*COS2-AV*AV
      B=AB*COS2-AV*BV
      C=B2*COS2-BV*BV
      TERM=B*B-A*C
      IF (TERM.LT.0.) GO TO 110
      IF (A.LE.0..AND.B.GT.0.) GO TO 110
      IF (C) 20,50,20
20    FL=C/(B-SQRT(TERM))
      IF (IPRINT.GE.2) WRITE (IOU6,170) COS2,BV,AV,U,V,W,A,B,C,TERM,FL,
     1A2,AB
      IF (LOT(I).NE.0.OR.(FL.GE.0..AND.FL.LE..7)) GO TO 30
      IF (KPRINT(3).GT.0) WRITE (IOU6,130) I,FL
      LOT(I)=I
      IF (I.NE.4) STEP=-.1
30    IF(FL.GE.-0.1.AND.FL.LE.1.1) GO TO 50
      IF ((I.NE.4).OR.(KPRINT(3).GT.0)) WRITE (IOU6,190) I,FL
      IF (N.EQ.4.OR.(WINGST.LE.YO.AND.YO.LE.WINGST+1.)) GO TO 50
40    ITEST=ITEST+1
      IF (I.EQ.4) LOT(4)=5
      RETURN
50    XBS(I)=HUB(1)+FL*XD1
      YBS(I)=HUB(2)+FL*YD1
      ZBS(I)=HUB(3)+FL*ZD1
60    UBS(I)=HUB(4)+FL*(UH(I)-HUB(4))
      VBS(I)=HUB(5)+FL*(VH(I)-HUB(5))
      WBS(I)=HUB(6)+FL*(WH(I)-HUB(6))
      PBS(I)=HUB(7)+FL*(PH(I)-HUB(7))
      PPMBS(I)=HUB(8)+FL*(PPMH(I)-HUB(8))
      CALL NORM (UBS(I),VBS(I),WBS(I),UBS(I),VBS(I),WBS(I))
      AMU=ASIN(SQRT(GF(2)/((PPMBS(I)/PBS(I))**GF(7)-1.)))
      IF (IPRINT.GE.2) WRITE (IOU6,170) U,V,W,A,B,C,TERM,FL,AMU,PBS(I),
     1PPMBS(I),HUB(7),WH(I),HUB(8),PPMH(I)
      IF (J.EQ.1) GO TO 70
      IF (FL.GT.1.) GO TO 80
      IF (ABS(AMU-AMUO).LE.EPSLN) GO TO 80
70    AMUO=AMU
      WRITE (IOU6,150) I
      CALL EXIT
80    IF (UBS(I)**2+WBS(I)**2.GT.4.E-06) GO TO 90
```

```
      ILL=ILL+1
      RETURN
90    THBS(I)=ACOS(VBS(I))
      PSBS(I)=ATAN2(WBS(I),UBS(I))
      IF (PSBS(I).LT.-3.) PSBS(I)=PSBS(I)+6.2831853072
100   CONTINUE
      IF (N.EQ.4) LOT(4)=0
      RETURN
110   WRITE (IOU6,140) I,A,B,C,TERM
      IF (I.NE.4.OR.N.EQ.4) CALL EXIT
      GO TO 40
C
C
C
120   FORMAT (//24H FLOW SUBSONIC, PPMC,PC=,2G15.7)
130   FORMAT (11H BASE POINT,I2,13H VIOLATES CFL,F10.5)
140   FORMAT (31HOINTERSEXION FAILURE, BASEPT NO,I2,4E18.8)
150   FORMAT (43HOMACH CONE CONVERGENCE FAILURE IN BASEPT NO,I2)
160   FORMAT (19HO***** BASEPT *****)
170   FORMAT (8E15.7)
180   FORMAT (14HOBASE POINT NO,I2)
190   FORMAT (8H BASE PT,I2,13H EXTRAPOLATES,F10.5,16H, STEP DECREASED)
      END
```

```
      SUBROUTINE COEFS (IND)
C     IND = 1 FOR BODY, 2 FOR FIELD, 3 FOR SHOCK
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /CONT/ NPC,DYC,ILL,KURE,SINE,COSINE
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC,A4,B4,C4,TH,
     1PS,HUB(8)
      COMMON /BSPT/ XBS(4),YBS(4),ZBS(4),UBS(4),VBS(4),WBS(4),PBS(4),
     1PPMBS(4),THBS(4),PSBS(4)
      DIMENSION DX(4),DY(4),DZ(4),DTH(4),DPS(4),A(4),B(4),C(4),D(4)
10    IF (IPRINT.GE.2) WRITE (IOU6,80)
      N=4
      IF (IND.EQ.2) GO TO 20
      N=3
      DX(4)=XC-HUB(1)
      DY(4)=YC-HUB(2)
      DZ(4)=ZC-HUB(3)
      DTH(4)=THETAC-TH
      DPS(4)=PSIC-PS
20    AMUC=ASIN(SQRT(GF(2)/((PPMC/PC)**GF(7)-1.)))
      DO 30 I=1,N
      DX(I)=XC-XBS(I)
      DY(I)=YC-YBS(I)
      DZ(I)=ZC-ZBS(I)
      DTH(I)=THETAC-THBS(I)
      DPS(I)=PSIC-PSBS(I)
      IF (IPRINT.GE.2) WRITE (IOU6,90) DX(I),DY(I),DZ(I),DTH(I),DPS(I)
30    CONTINUE
      DO 50 I=1,N
      CALL NORM (.5*(UBS(I)+UC),.5*(VBS(I)+VC),.5*(WBS(I)+WC),UNT,VNT,
     1WNT)
      IF (IPRINT.GE.2) WRITE (IOU6,90) UNT,VNT,WNT
      J=MOD(I,4)+1
      K=MOD(I-5,4)+4
      CALL SOL (DX(I),DY(I),DZ(I),DTH(I),DX(J),DY(J),DZ(J),DTH(J),DX(K),
     1DY(K),DZ(K),DTH(K),DTHDX,DTHDY,DTHDZ)
      CALL SOL (DX(I),DY(I),DZ(I),DPS(I),DX(J),DY(J),DZ(J),DPS(J),DX(K),
     1DY(K),DZ(K),DPS(K),DPSDX,DPSDY,DPSDZ)
      CALL NORM (VNT*DZ(I)-WNT*DY(I),WNT*DX(I)-UNT*DZ(I),UNT*DY(I)-VNT*
     1DX(I),DXDN,DYDN,DZDN)
```

```
      DTHDN=DTHDX*DXDN+DTHDY*DYDN+DTHDZ*DZDN
      DPSDN=DPSDX*DXDN+DPSDY*DYDN+DPSDZ*DZDN
      IF (IPRINT.GE.2) WRITE (IOU6,90) DXDN,DYDN,DZDN,DTHDN,DPSDN
      VCHEK=SQRT(UNT*UNT+WNT*WNT)
      IF (VCHEK.GT.2.E-3) GO TO 40
      ILL=ILL+1
      RETURN
40    COSD1=(UNT*DZDN-WNT*DXDN)/VCHEK
      IF (ABS(COSD1).GT.1.) COSD1=SIGN(1.,COSD1)
      UZWX=UNT*DZ(I)-WNT*DX(I)
      SIND1=SIGN(SQRT(1.-COSD1*COSD1),UZWX)
      AMUAV=.5*(AMUC+ASIN(SQRT(GF(2)/((PPMBS(I)/PBS(I))**GF(7)-1.))))
      SINMU=SIN(AMUAV)
      SINTH=SIN(.5*(THETAC+THBS(I)))
      DL=SQRT(DX(I)*DX(I)+DY(I)*DY(I)+DZ(I)*DZ(I))
      IF (IPRINT.GE.2) WRITE (IOU6,90) AMUAV,SINMU,SINTH,SIND1,COSD1,DL
      A(I)=SINMU*COS(AMUAV)/(GF(1)*.5*(PC+PBS(I)))
      B(I)=-COSD1
      C(I)=SINTH*SIND1
      D(I)=A(I)*PBS(I)+B(I)*THBS(I)+C(I)*PSBS(I)-DL*SINMU*(SIND1*DTHDN+
     1SINTH*COSD1*DPSDN)
      IF (IPRINT.GE.2) WRITE (IOU6,90) A(I),B(I),C(I),D(I)
50    CONTINUE
      DO 60 I=1,9
60    COF(I)=0.
      DO 70 I=1,N
      COF(1)=COF(1)+A(I)*D(I)
      COF(2)=COF(2)+B(I)*D(I)
      COF(3)=COF(3)+C(I)*D(I)
      COF(4)=COF(4)+A(I)*A(I)
      COF(5)=COF(5)+A(I)*B(I)
      COF(6)=COF(6)+A(I)*C(I)
      COF(7)=COF(7)+B(I)*B(I)
      COF(8)=COF(8)+B(I)*C(I)
70    COF(9)=COF(9)+C(I)*C(I)
      IF (IPRINT.GE.2) WRITE (IOU6,90) (COF(I),I=1,9)
      RETURN
C
C
C
```

```
80     FORMAT (18H0***** COEFS *****)                              E  81
90     FORMAT (8E15.7)                                             E  82
       END                                                         E  83
```

```
      SUBROUTINE COMPAT
C**** Y-AXIS MUST ALIGN WITH FREE STREAM, ROTATE AXES IF NEEDED
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP,COF(9)
      COMMON /ESENS/ XC,YC,ZC,UC,VC,WC,PC,PPMC,THETAC,PSIC
      IF (IPRINT.GE.2) WRITE (IOU6,40)
      GM2=GF(1)*FMSTR*FMSTR
      DO 20 I=1,20
      XI=PC*POP
      D1=XI-1.
      D2=GM2-D1
      D4=XI+GF(4)
      D3=GM2/GF(3)-D4
      THETAC=ATAN(SQRT(D3/D4)*D1/D2)
      PSIC=(COF(3)-COF(8)*THETAC-COF(6)*PC)/COF(9)
      SN2T=.5*SIN(2.*THETAC)
      DTHDP=SN2T*(1./D1+1./D2-.5/D3-.5/D4)*POP
      DPSDP=-(COF(8)*DTHDP+COF(6))/COF(9)
      D2TDP2=COS(2.*THETAC)/SN2T*DTHDP*DTHDP+POP**2*SN2T*(-1./D1**2+1./
     1D2**2-.5/D3**2+.5/D4**2)
      C=COF(5)*PC+COF(7)*THETAC+COF(8)*PSIC-COF(2)
      H=C*DTHDP+COF(4)*PC+COF(5)*THETAC+COF(6)*PSIC-COF(1)
      HP=C*D2TDP2+(COF(7)*DTHDP+COF(8)*DPSDP+2.*COF(5))*DTHDP+COF(6)*
     1DPSDP+COF(4)
      PC=PC-H/HP
      IF (IPRINT.GE.2) WRITE (IOU6,50) PC,THETAC,PSIC,XI,D1,D2,D3,D4,
     1SN2T,DTHDP,DPSDP,D2TDP2,C,H,HP
      IF (I.EQ.1) GO TO 10
      IF (ABS((PC-PCN)/PC).GT.DP) GO TO 10
      IF (ABS(THETAC-THCN).GT.DTHETA) GO TO 10
      IF (ABS(PSIC-PSCN).LE.DPSI) GO TO 30
10    PCN=PC
      THCN=THETAC
20    PSCN=PSIC
      WRITE (IOU6,60)
      CALL EXIT
30    PPMC=(D4/(GF(4)*XI+1.))**GF(6)/XI**GF(5)
      RETURN
C
C
```

```
C
40    FORMAT (19H0***** COMPAT *****)
50    FORMAT (8E15.7)
60    FORMAT (30H0CONVERGENCE FAILURE IN COMPAT)
      END
```

```
      SUBROUTINE GMTRY (SINE,COSINE,XC,YC,ZC,A4,B4,C4)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /GEOM/ YMAX,YSTOP,UNIL,INDEX,ZBLNU(4),YBLNU(4),DBLNL
      UNIL=AMAX1(1.,UNIL)
      X1=COSINE*XC-SINE*YC
      YC=SINE*XC+COSINE*YC
      TA=COSINE*A4-SINE*B4
      TB=SINE*A4+COSINE*B4
      XC=X1
      Z1=ZC
      IND=0
      DO 40 J=1,5
      INDEX=0
      IN=IND
      T=0.
      DO 30 I=1,30
      CALL FIGURE (XC,YC,ZC,F,A,B,C)
      IF (INDEX.EQ.IND) GO TO 50
      IF (I.GT.1) GO TO 10
      B1=TA*B-TB*A
      B3=C4*B-TB*C
10    T=T-F/(B1*A+B3*C)
      XC=X1+T*B1
      ZC=Z1+T*B3
      IF (IPRINT.GE.2) WRITE (IOU6,60)
      IF (IPRINT.GE.2) WRITE (IOU6,70) T,XC,YC,ZC,F,A,B,C,INDEX,IND,J
      IF (I.EQ.1) GO TO 20
      IF (ABS(XC-XO)+ABS(ZC-ZO).LE.UNIL*1.E-6) GO TO 40
20    XO=XC
      ZO=ZC
30    CONTINUE
      WRITE (IOU6,80)
      CALL EXIT
40    IND=INDEX
      WRITE (IOU6,90) INDEX,IN
50    CALL ROTATE (XC,YC,SINE,COSINE)
      CALL NORM (COSINE*A+SINE*B,-SINE*A+COSINE*B,C,A4,B4,C4)
      RETURN
C
C
```

```
C
60    FORMAT (25(2H *),6H GMTRY,25(2H *))
70    FORMAT (8G15.7,3I3)
80    FORMAT (29H0CONVERGENCE FAILURE IN GMTRY)
90    FORMAT (40H0BODY POINT OSCILLATING BETWEEN SECTIONS,I2,4H AND,I2)
      END
```

```
      SUBROUTINE FIGURE (X,Y,Z,F,A,B,C)
      COMMON /MESH/ K1(8),AL,ZZL,ZX23,ZX7,SP,XXL
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      COMMON /GEOM/ YMAX,YSTOP,UNIL,INDEX,DBLNL,NCQ,CQ(8,10)
      DIMENSION ZX(23),ZXP(23),P(23),Q(23),R(23),S(23),T(23),SG(23)
      DIMENSION AU(2),SU(3)
      IF ((Y.EQ.YI).AND.(MOLD.LT.3)) GO TO 80
      IF ((Y.EQ.YI).AND.(MOLD.GT.2)) GO TO 70
      YI=Y
      IF ((Y.GT.YSTOP+.0001).AND.(ABS(YSTOP).GT.EPSLN)) GO TO 380
      DO 30 I=1,50
      IF (Y.LE.YMAX) GO TO 40
      READ (IOU5,400) YMAX,N,MOD,RATIO
      IF (EOF(IOU5)) 380,10
10    IF (MOD.NE.0) MOLD=MOD
      WRITE (IOU6,440) YMAX,N,MOLD,RATIO
      DO 20 K=1,N
      READ (IOU5,420) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
      WRITE (IOU6,450) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
20    CONTINUE
      IF (MOLD.EQ.1) M=7
      IF (MOLD.EQ.2) M=13
      IF (MOLD.GT.2) M=21
30    CONTINUE
40    DO 50 I=1,M
      IF ((RATIO.GT.0.).AND.((I.EQ.19).OR.(I.EQ.20))) GO TO 50
      IF (MOLD.EQ.3.AND.(I.GT.7.AND.I.LT.14)) GO TO 50
      CALL CONICF (P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
      IF (IPRINT.GE.2) WRITE (IOU6,390) I
50    CONTINUE
      AL=ZX(18)
      IF (MOLD.LT.3) ZX7 = ZX(7)
      IF (MOLD.LT.3) GO TO 80
      IF (RATIO.GT.0.) CALL XSUBG (ZX,ZXP,RATIO)
      CALL CONPIC (CQ,NCQ,Y,ZX(22),ZXP(22),INQ)
      IF (INDEX.LT.-1) GO TO 60
      ZX(23)=ZX(15)-DBLNL
      ZXP(23)=ZXP(15)
      ZX23=ZX(23)
      UNIL=ABS(ZX(5))
```

```
      ZX7=AMIN1(ZX(23),ZX(7))
60    CALL PCRCL (ZX,ZXP,-2.,Y,Z,14,16,19,1,4,2,5,22,A,B,C,F,X2U,Z2U)
      CALL PCRCL (ZX,ZXP,-2.,Y,Z,15,17,20,3,6,7,5,23,A,B,C,F,X2L,Z2L)
      IF (INDEX.EQ.-1) XXL=X2L
      IF (INDEX.EQ.-1) ZZL=Z2L
      CALL ROTXY (X2U,Z2U,X2U,Z2U,ZX(19),ZX(16),X2UP,Z2UP)
      CALL ROTXY (ZX(19),ZX(16),X2U,Z2U,ZX(19),ZX(16),ZX19,ZX16)
      CALL ROTXY (X2L,Z2L,X2L,Z2L,ZX(20),ZX(17),X2LP,Z2LP)
      CALL ROTXY (ZX(20),ZX(17),X2L,Z2L,ZX(20),ZX(17),ZX20,ZX17)
      IF (INDEX.LT.0) RETURN
70    CALL ROTXY (X,Z,X2U,Z2U,ZX(19),ZX(16),XUP,ZUP)
      CALL ROTXY (X,Z,X2L,Z2L,ZX(20),ZX(17),XLP,ZLP)
80    IF (INDEX.GT.0) GO TO (140,160,180,200,220,240,260,280,300,320,340
     1,360,120), INDEX
      IF ((MOLD.EQ.1).OR.(MOLD.EQ.3)) GO TO 90
      IF ((Z.GE.ZX(10)).AND.(X.LE.ZX(13))) GO TO 130
      IF (Z.GE.ZX(8)) GO TO 150
      IF (Z.GT.ZX(9)) GO TO 170
90    IF ((MOLD.LT.3).AND.(Z.GT.ZX(2))) GO TO 190
      IF ((MOLD.GT.2).AND.(Z.GT.ZX(22)).AND.(Z.GT.ZX(2)).AND.(X.LT.X2U)
     1.AND.(ZX(14).LE.ZX(16))) GO TO 190
      IF ((MOLD.GT.2).AND.(Z.GT.ZX(22)).AND.(Z.GT.ZX(2)).AND.(ZX(14)
     1.GT.ZX(16))) GO TO 190
      IF (MOLD.LT.3) GO TO 110
      IF ((Z.GE.ZX(22)).AND.(X.LT.X2U).AND.(ZX(14).LE.ZX(16))) GO TO 210
      IF ((Z.GE.ZX(22)).AND.(ZX(14).GT.ZX(16))) GO TO 210
      IF (Z.LT.ZX(18)) GO TO 100
      IF (XUP.LT.X2UP) GO TO 230
      IF (XUP.LT.ZX19) GO TO 250
      GO TO 270
100   IF ((Z.LE.ZX(23)).AND.(X.LT.X2L).AND.(ZX(15).GE.ZX(17))) GO TO 110
      IF ((Z.LE.ZX(23)).AND.(ZX(15).LT.ZX(17))) GO TO 110
      IF (XLP.LT.X2LP) GO TO 290
      IF (XLP.LT.ZX20) GO TO 310
      GO TO 330
110   IF (Z.GE.ZX(7)) GO TO 210
      IF (X.GT.ZX(6)) GO TO 350
C****BOTTOM FLAT SECTION
      INDEX=13
120   A=0.0
```

```
      B=ZXP(3)
      C=-1.0
      F=-(Z-ZX(3))
      GO TO 370
C****CANOPY FLAT SECTION
130   INDEX=1
140   A=0.0
      B=-ZXP(10)
      C=1.0
      F=Z-ZX(10)
      GO TO 370
C****CANOPY SECTION
150   INDEX=2
160   CALL ELIPSE (X,Z,ZX,ZXP,10,11,13,12,A,B,C,F)
      GO TO 370
C****CUBIC SECTION
170   INDEX=3
180   CALL CUBIC (X,Y,Z,ZX,ZXP,A,B,C,F)
      GO TO 370
C****UPPER FUSELAGE ELLIPSE SECTION
190   INDEX=4
200   CALL ELIPSE (X,Z,ZX,ZXP,1,2,4,5,A,B,C,F)
      GO TO 370
C****VERTICAL STRAIGHT SECTION
210   INDEX=5
220   A=1.0
      B=-ZXP(5)
      C=0.
      F=X-ZX(5)
      GO TO 370
C****UPPER WING BLENDING SECTION
230   INDEX=6
240   CALL PCRCL (ZX,ZXP,X,Y,Z,14,16,19,1,4,2,5,22,A,B,C,F,D,D)
      GO TO 370
C****UPPER WING STRAIGHT SECTION
250   INDEX=7
260   CALL WINGF (ZX,ZXP,X,Y,Z,19,5,16,14,A,B,C,F)
      GO TO 370
C****UPPER WING TIP SECTION
270   INDEX=8
```

```
280   CALL PELIPS (ZX,ZXP,X,Y,Z,14,16,19,5,18,21,A,B,C,F)
      GO TO 370
C****LOWER WING BLENDING SECTION
290   INDEX=9
300   CALL PCRCL (ZX,ZXP,X,Y,Z,15,17,20,3,6,7,5,23,A,B,C,F,D,D)
      GO TO 370
C****LOWER WING STRAIGHT SECTION
310   INDEX=10
320   CALL WINGF (ZX,ZXP,X,Y,Z,20,5,17,15,A,B,C,F)
      GO TO 370
C****LOWER WING TIP SECTION
330   INDEX=11
340   CALL PELIPS (ZX,ZXP,X,Y,Z,15,17,20,5,18,21,A,B,C,F)
      GO TO 370
C****LOWER FUSELAGR ELLIPSE SECTION
350   INDEX=12
360   CALL ELIPSE (X,Z,ZX,ZXP,3,7,6,5,A,B,C,F)
370   IF (IPRINT.GE.2) WRITE (IOU6,410) F,A,B,C,INDEX
      RETURN
380   WRITE (IOU6,430)
      CALL EXIT
C
C
390   FORMAT (10H CURVE NO ,I3,5H   OK)
400   FORMAT (F10.5,2I5,F10.5)
410   FORMAT (19H0***** FIGURE *****/4G14.7,21H BODY SECTION INDEX =,I2)
420   FORMAT (I2,5E15.8,F2.0)
430   FORMAT (1H0,60(2H *)/1H0,20X,35H$$$$$ HALLELUJAH, YOU HAVE GONE AL
     1L,35H THE WAY--FEELING NICE, I BET $$$$$/1H0,60(2H *))
440   FORMAT (35H0GEOMETRIC COEFFICIENTS FOR Y UP TO,G15.7,I5,11H NEW CU
     1RVES,2X,5HMOLD=,I2,2X,6HRATIO=,F7.3)
450   FORMAT (9H0CURVE NO,I3,5H, P =E16.8,5H  Q =E16.8,5H  R =E16.8,5H
     1S =E16.8,5H  T =E16.8,6H  SG =F4.1)
      END
```

```
SUBROUTINE ROTXY (X,Y,X1,Y1,X2,Y2,XP,YP)
DY=Y2-Y1
DX=X2-X1
R=SQRT(DX**2+DY**2)
SINP=DY/R
COSP=DX/R
XP=COSP*X+SINP*Y
YP=-SINP*X+COSP*Y
RETURN
END
```

```
      SUBROUTINE TRANS (H,XYZ,A4,B4,C4,TO,IN)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      DIMENSION H(4,8),XYZ(3,7)
      IF (A4*A4+C4*C4.LE.1.E-14) RETURN
      COSX=-SQRT(1.-C4*C4)
      SRT=SQRT(A4*A4+B4*B4)
      IF (SRT.GT.1.E-7) GO TO 10
      COSZ=0.
      SINZ=1.*TO
      GO TO 20
10    COSZ=-B4/SRT
      SINZ=TO*A4/SRT
20    DO 50 J=1,7
      IF (IN.EQ.0.AND.J.GT.4) GO TO 50
      IF (IN.NE.0.AND.J.GT.3) GO TO 40
      DO 30 K=1,4,3
      IF (TO.GT.0.) CALL ROTATE (H(J,K),H(J,K+1),SINZ,COSZ)
      CALL ROTATE (H(J,K+1),H(J,K+2),C4*TO,COSX)
      IF (TO.LT.0.) CALL ROTATE (H(J,K),H(J,K+1),SINZ,COSZ)
30    CONTINUE
      IF (IN.EQ.0) GO TO 50
40    IF (J.GT.2.AND.J.LT.6) GO TO 50
      IF (TO.GT.0.) CALL ROTATE (XYZ(1,J),XYZ(2,J),SINZ,COSZ)
      CALL ROTATE (XYZ(2,J),XYZ(3,J),C4*TO,COSX)
      IF (TO.LT.0.) CALL ROTATE (XYZ(1,J),XYZ(2,J),SINZ,COSZ)
50    CONTINUE
      IF (IPRINT.GE.2) WRITE (IOU6,60) H,XYZ
      IF (TO.LT.0..OR.IN.EQ.0) RETURN
      XYZ(3,3)=ACOS(XYZ(2,2))
      XYZ(1,4)=ATAN2(XYZ(3,2),XYZ(1,2))
      XYZ(2,5)=ACOS(XYZ(2,7))
      XYZ(3,5)=ATAN2(XYZ(3,7),XYZ(1,7))
      IF (XYZ(1,4).LT.-3.) XYZ(1,4)=XYZ(1,4)+6.2831853072
      IF (XYZ(3,5).LT.-3.) XYZ(3,5)=XYZ(3,5)+6.2831853072
      RETURN
C
C
C
60    FORMAT (18HO***** TRANS *****/(8E15.7))
      END
```

```
SUBROUTINE ROTATE (X,Y,SINE,COSINE)
TEMP=X
X=COSINE*TEMP+SINE*Y
Y=-SINE*TEMP+COSINE*Y
RETURN
END
```

```
      SUBROUTINE SOL (A11,A12,A13,D1,A21,A22,A23,D2,A31,A32,A33,D3,X,Y,Z
     1)
      IF (ABS(A11).LE.1.E-7) GO TO 10
      B12=A12/A11
      B13=A13/A11
      C1=D1/A11
      B22=A22-B12*A21
      IF (ABS(B22).LE.1.E-7) GO TO 10
      B32=A32-B12*A31
      B23=(A23-B13*A21)/B22
      C2=(D2-A21*C1)/B22
      B33=(A33-A31*B13-B23*B32)
      IF (ABS(B33).LE.1.E-7) GO TC 10
      Z=(D3-C1*A31-C2*B32)/B33
      Y=C2-B23*Z
      X=C1-B12*Y-B13*Z
      RETURN
10    D=A11*(A22*A33-A23*A32)+A21*(A13*A32-A12*A33)+A31*(A12*A23-A13*A22
     1)
      X=(A13*(D2*A32-A22*D3)+A23*(A12*D3-D1*A32)+A33*(D1*A22-A12*D2))/D
      Y=(A11*(D2*A33-A23*D3)+A21*(A13*D3-D1*A33)+A31*(D1*A23-A13*D2))/D
      Z=(A12*(D2*A31-A21*D3)+A22*(A11*D3-D1*A31)+A32*(D1*A21-A11*D2))/D
      RETURN
      END
```

```
      SUBROUTINE HARNES (QINF,A,N)
      DIMENSION A(99,8,4)
      DO 10 L=3,N
      VN=SQRT(A(L,4)**2+(QINF-A(L,5))**2+A(L,6)**2)
      CALL NORM (-A(L,4),QINF-A(L,5),-A(L,6),A0,B0,C0)
      A1=(A(L+1,3)-A(L,3))**2+(A(L+1,1)-A(L,1))**2
      C1=(A(L,3)-A(L-1,3))**2+(A(L,1)-A(L-1,1))**2
      CALL NORM (C1*(A(L+1,3)-A(L,3))+A1*(A(L,3)-A(L-1,3)),0.,C1*(A(L+1,
     11)-A(L,1))+A1*(A(L,1)-A(L-1,1)),A1,B1,C1)
      B=SQRT(1.-B0*B0)
      CALL NORM (SIGN(A1*B,A0)+A0,0.,SIGN(C1*B,C0)+C0,A1,B1,C1)
      A(L,4)=-A1*VN*B
   10 A(L,6)=-C1*VN*B
      RETURN
      END
```

```
      SUBROUTINE MIRROR (A,SIN2A,COS2A,N)
      DIMENSION A(99,8)
      I=1
      J=3
      DO 20 L=1,2
      DO 10 K=1,4,3
      A(I,K)=-COS2A*A(J,K)+SIN2A*A(J,K+1)
      A(I,K+1)=SIN2A*A(J,K)+COS2A*A(J,K+1)
      A(I,K+2)=A(J,K+2)
   10 A(I,K/4+7)=A(J,K/4+7)
      I=N+L
   20 J=N-L
      RETURN
      END
```

```
SUBROUTINE NORM (X,Y,Z,A,B,C)
T=SQRT(X*X+Y*Y+Z*Z)
A=X/T
B=Y/T
C=Z/T
RETURN
END
```

```
      SUBROUTINE DATOUT (A,I,IND,KAB,IO)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3
      COMMON /FSTR/ COSA,SINA,FMSTR,QINF,GF(7),POP
      COMMON /CONT/ NPC,DVC,ILL,KURE,SINE,COSINE,N,Y,IS
      DIMENSION A(99,8)
      IF (N.EQ.IS.AND.IO.NE.0) RETURN
      IF (KP4.LE.0.AND.(IO.EQ.1.AND.N/KAB*KAB.NE.N)) RETURN
      IF ((KP4.GT.0.AND.IO.EQ.0).AND.(N/KAB*KAB.NE.N.AND.(IND.NE.1.OR.(
     1I.NE.2.AND.I.NE.NPC+1)))) RETURN
      RAD=57.2957795130823
      P=A(I,7)*POP
      P1=A(I,7)/A(I,8)
      CP=2.*(P-1.)/(GF(1)*FMSTR*FMSTR)
      XM=SQRT((1./P1**GF(7)-1.)/GF(2))
      X=COSINE*A(I,1)-SINE*A(I,2)
      U=COSINE*A(I,4)-SINE*A(I,5)
      V=SINE*A(I,4)+COSINE*A(I,5)
      OW=ATAN(U/V)*RAD
      UW=ATAN(A(I,6)/V)*RAD
      PS=0.
      TH=0.
      IF (ABS(A(I,6)).LE.1.E-07.AND.ABS(U).LE.1.E-07) GO TO 10
      PS=ATAN2(A(I,6),U)*RAD
      TH=ACOS(V/SQRT(U*U+V*V+A(I,6)*A(I,6)))*RAD
10    IF (IO.EQ.0.OR.KP4.GE.0) GO TO 20
      WRITE (IOU7,120) A(I,3),X,XM,A(I,8),TH,PS
      RETURN
20    IOU=IOU6
      IF (KP4.GE.0.AND.IO.EQ.1) IOU=IOU3
      IF (I.GT.2) GO TO 30
      Y=SINE*A(I,1)+COSINE*A(I,2)
      IF (IND.EQ.1) WRITE (IOU,110) N,Y
      IF (KP4.GT.1.AND.(IO.EQ.0.AND.N/KAB*KAB.NE.N)) GO TO 40
      IF (IND.EQ.1) WRITE (IOU,60)
      IF (IND.EQ.2) WRITE (IOU,70)
      IF (IND.EQ.3) WRITE (IOU,80)
      WRITE (IOU,90)
30    IF (KP4.GT.1.AND.(IO.EQ.0.AND.N/KAB*KAB.NE.N)) GO TO 50
      WRITE (IOU,100) A(I,3),X,XM,A(I,8),TH,PS,CP,P,A(I,7),OW,UW
      RETURN
```

```
40    WRITE (IOU6,130) CP
50    IF (I.EQ.NPC+1) WRITE (IOU6,140) CP
      RETURN
C
C
C
60    FORMAT (1H054X11HBODY POINTS)
70    FORMAT (1H054X12HFIELD POINTS)
80    FORMAT (1H054X12HSHOCK POINTS)
90    FORMAT (80H ------Z-------  ------X------  MACH NO.  ----PT/PTO---
     1- THETA, DEG  --PSI,DEG--,52H  -----CP----- ----P/PO---- ---P/PTO-
     2--OUTWASHUPWASH)
100   FORMAT (1H 2E15.8,F10.7,E15.8,F12.8,F13.8,3E13.5,2F6.2)
110   FORMAT (11H0SURFACE NO,I3,14H, STATION Y = E15.8)
120   FORMAT (2E15.8,F10.7,E15.8,F12.8,F13.8)
130   FORMAT (1H+,44X,22HLOWER CENTERLINE CP = E15.8)
140   FORMAT (1H+,83X,22HUPPER CENTERLINE CP = E15.8)
      END
```

```
      SUBROUTINE TIDYUP (SINE,COSINE,A,NP,NPP,AN,LINE)
      COMMON /MESH/ K1,K2,K3,K4,K5,K6,K7,K8,ZL(6)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI
      DIMENSION A(99,8,4),L(5),TM(5),ZT(99),XT(99)
      IF (K5.EQ.0) ZL(1)=AN
      DO 10 I=1,NP
      IF (I.LE.5) L(I)=0
      CALL ROTATE (A(I,1,1),A(I,2,1),-SINE,COSINE)
   10 CALL ROTATE (A(I,4,1),A(I,5,1),-SINE,COSINE)
      A(1,1,4)=0.
      REFL=1.E10
      IF (LINE.EQ.1) GO TO 30
      NO=NP-K7
      DO 20 I=1,NO
      XT(I)=A(I+K7,1,1)
      ZT(I)=A(I+K7,3,1)
   20 CALL ROTATE (XT(I),ZT(I),-.7071067812,.7071067812)
      XP=XO
      CALL SECOND (XT,ZT,NO,XP,ZL(5),DYP,0)
      CALL ROTATE (XP,ZL(5),.7071067812,.7071067812)
   30 DO 60 I=2,NP
      DO 50 KK=1,5
      IF (L(KK).NE.0.OR.KK.EQ.2) GO TO 40
      IF (A(I,3,1).LT.ZL(KK).AND.A(I+1,3,1).GE.ZL(KK)) L(KK)=I
   40 IF (REFL.LT.(A(I,3,1)-ZL(2))**2+(A(I,1,1)-ZL(6))**2) GO TO 50
      L(2)=I
      REFL=(A(I,3,1)-ZL(2))**2+(A(I,1,1)-ZL(6))**2
   50 CONTINUE
   60 A(I,1,4)=A(I-1,1,4)+SQRT((A(I,1,1)-A(I-1,1,1))**2+(A(I,2,1)-A(I-1,
     12,1))**2+(A(I,3,1)-A(I-1,3,1))**2)
      DO 70 I=1,5
      IF (LINE.EQ.1.AND.I.EQ.5) GO TO 80
      IF (K5.EQ.C.AND.I.GT.1) GO TO 80
   70 TM(I)=(A(L(I)+1,1,4)*(ZL(I)-A(L(I),3,1))+A(L(I),1,4)*(A(L(I)+1,3,1
     1)-ZL(I)))/(A(L(I)+1,3,1)-A(L(I),3,1))
   80 NT=2*NPP-6+K1-K2+2*(K3-K4)+4*(K4-K5+K6-K7)-2*(K4-K5+K6-K7)/(K8+1)
      IF (K6-K5.LT.2) NT=NT+4+2*(K5-K6)
      IF (K5.EQ.0) NT=2*NPP-2+K1-K2+K3-K4
      K9=2*K7-K6
      K10=2*K7-K5
```

```
      K11=2*K7-K4
      K12=2*K7-K3
      N=0
      DO 90 M=2,NPP
      N=N+1
      IF (M.LE.K1.OR.M.GT.K2) N=N+1
      IF (N.GE.NT/2) GO TO 100
90    CONTINUE
100   IF (K5.NE.0) M=K12
      IF (K5.NE.0) N=(NT-K1+K2)/2+K3-1+K12-NPP
      DTO=TM(1)/FLOAT(N)
      IF (LINE.EQ.1) RATIO=TM(4)/TM(1)
      IF (K5.NE.C) DTO5=TM(1)*(1.-RATIO)/(FLOAT(N-2*K3))
      IF (K5.NE.C) DTO=(TM(1)*RATIO-2.*DTO5)/FLOAT(2*K3-2)
      DT1=(A(NP,1,4)-TM(1))/FLOAT(NT-N)
      DTO7=(TM(5)-TM(1))/FLOAT(N-2*K3+2)
      TOP=2*(NPP-K12)+K1-K2
      IF (K5.NE.0.AND.K5.EQ.K6) DT1=DTO5+(DT1-DTO5)*FLOAT(NT-N)/TOP
      IF (K5.NE.K6.AND.LINE.NE.1) DT1=DTO7+(DT1-DTO7)*FLOAT(NT-N)/TOP
      IF (K3.NE.0.AND.K5.EQ.0) DTO=A(NP,1,4)/FLOAT(NT)
      IF (K3.NE.0.AND.K5.EQ.0) DT1=DTO
      IF (K5.EQ.K6) K=0
      IF (K5.EQ.K6.OR.LINE.NE.1) GO TO 110
      DT1=DTO5+(DT1-DTO5)*FLOAT(NT-N)/TOP
      SHIFT=DTO5*(FLOAT(K4-K3)+FLOAT((K5-K4)/(K8+1))/2.)
      MOVE=((TM(2)+TM(3))/2.-SHIFT-DTO*FLOAT(2*K3-2))*FLOAT(K8+1)/DTO5
      MOVE=MAXO(MOVE,1-(K4-K3)*(K8+1))
      K=MOVE/(K8+1)
      IF (MOVE.LT.0) K=K-1
      INC=MOVE-K*(K8+1)-1
      IF (MOVE.EQ.0) INC=0
      IF (MOVE.GE.4) INC=INC+1
      IF (IPRINT.GE.2) WRITE (IOU6,150) MOVE,K,INC
110   DO 130 I=1,8
      DT=DTO
      LL=1
      IF (I.EQ.1.OR.I.EQ.4) LL=2
      CALL KURFIT (A(1,1,4),A(1,I,1),A(1,1,3),NP,0.,0.,LL,LL)
      PT=-DT-DT
      DO 130 J=1,NPP
```

```
      IF (K5.NE.0.AND.J.GT.K3) DT=DT05
      IF (K5.NE.K6.AND.(LINE.NE.1.AND.J.GT.K7)) DT=DT07
      IF (J.GT.M) DT=DT1
      PT=PT+DT
      IF (J.LE.K1.OR.J.GT.K2) PT=PT+DT
      IF ((J.GT.K3.AND.J.LE.K4).OR.(J.GT.K4.AND.J.LE.K12)) PT=PT-DT
      IF ((J.GT.K5+1.AND.J.LT.K6).OR.(J.GT.K9+1.AND.J.LT.K10)) PT=PT+DT
      IF ((J.GT.K4+K.AND.J.LE.K5+K).OR.((J.GT.K6.AND.J.LE.K9).OR.(
     1J.GT.K10-K.AND.J.LE.K11-K))) PT=PT-DT*FLOAT(K8)/FLOAT(K8+1)
      IF (K5.EQ.K6) GO TO 120
      IF (J.EQ.K4+K+1.OR.J.EQ.K11-K) PT=PT+FLOAT(INC)*DT/FLOAT(K8+1)
      IF (J.EQ.K5+K+1.OR.J.EQ.K10-K) PT=PT-FLOAT(INC)*DT/FLOAT(K8+1)
120   CALL KURVE (A(1,1,3),A(1,1,4),A(1,I,1),PT,A(J,I,2),NP,LL)
130   CONTINUE
      IF (LINE.EQ.1) XO=A(M,1,2)
      IF (LINE.EQ.1) ZO=A(M,3,2)
      IF (LINE.EQ.1) CALL ROTATE (XO,ZO,-.7071067812,.7071067812)
      IF (IPRINT.GE.2) WRITE (IOU6,160) ((A(I,J,2),J=1,8),I=1,NPP)
      DO 140 I=1,NPP
      CALL ROTATE (A(I,1,2),A(I,2,2),SINE,COSINE)
140   CALL ROTATE (A(I,4,2),A(I,5,2),SINE,COSINE)
      RETURN
C
C
C
150   FORMAT (6I10)
160   FORMAT (19H0***** TIDYUP *****/(8E14.6))
      END
```

```
      SUBROUTINE SECOND (X,Y,N,XP,YP,DYP,IT)
      DIMENSION X(1),Y(1)
      M=N-1
      ARROW=XP-X(1)
      NI=N/2+1
      IF ((X(NI)-X(1))*ARROW) 10,10,20
10    IP=2
      GO TO 40
20    DO 30 I=2,M
      IP=I
      IF ((XP-X(I))*ARROW) 40,40,30
30    CONTINUE
40    A=XP-X(IP-1)
      B=XP-X(IP)
      C=XP-X(IP+1)
      D=X(IP-1)-X(IP)
      E=X(IP-1)-X(IP+1)
      G=X(IP)-X(IP+1)
      T1=Y(IP-1)/(D*E)
      T2=-Y(IP)/(D*G)
      T3=Y(IP+1)/(E*G)
      YP=T1*B*C+T2*A*C+T3*A*B
      IF (IT.GT.0) DYP=T1*(B+C)+T2*(A+C)+T3*(A+B)
      RETURN
      END
```

```
      SUBROUTINE REFORM (A,LTAPE,NTAPE,NPC,LINE,NLINE)
      DIMENSION A(99,8,4),X(3)
      REWIND LTAPE
      ALINE=NLINE-1
      DO 80 I=1,NPC
      REWIND NTAPE
      DO 40 J=1,LINE
      READ (NTAPE) ((A(L,K,2),K=1,8),L=1,NPC)
      IF (I.NE.1.OR.J.NE.1) GO TO 20
      DO 10 K=1,3
10    X(K)=.5*(A(1,K,2)+A(NPC,K,2))
20    DO 30 K=1,8
30    A(J,K)=A(I,K,2)
40    A(J,8)=ALOG10(-ALOG10(A(J,8)))
      A(1,1,4)=SQRT((A(1,1)-X(1))**2+(A(1,2)-X(2))**2+(A(1,3)-X(3))**2)
      DO 50 J=2,LINE
50    A(J,1,4)=A(J-1,1,4)+SQRT((A(J,1)-A(J-1,1))**2+(A(J,2)-A(J-1,2))**2
     1+(A(J,3)-A(J-1,3))**2)
      DA=(A(LINE,1,4)-A(1,1,4))/ALINE
      DO 60 K=1,8
      CALL KURFIT (A(1,1,4),A(1,K),A(1,1,3),LINE,0.,0.,2,2)
      DO 60 J=1,NLINE
      PA=A(1,1,4)+DA*FLOAT(J-1)
60    CALL KURVE (A(1,1,3),A(1,1,4),A(1,K),PA,A(J,K,2),LINE,K)
      DO 70 J=1,NLINE
70    A(J,8,2)=10.**(-10.**A(J,8,2))
80    WRITE (LTAPE) ((A(M,K,2),K=1,8),M=1,NLINE)
      REWIND NTAPE
      DO 100 J=1,NLINE
      REWIND LTAPE
      DO 90 I=1,NPC
      READ (LTAPE) ((A(M,K,2),K=1,8),M=1,NLINE)
      DO 90 K=1,8
90    A(I,K)=A(J,K,2)
100   WRITE (NTAPE) ((A(L,K),K=1,8),L=1,NPC)
      LINE=NLINE
      RETURN
      END
```

```
      SUBROUTINE KURFIT (X,Y,A,N,DY1,DY2,K1,K2)
      DIMENSION X(1),Y(1),A(1),B(200)
      N1=N-2
      A(1)=-DY1/2.
      B(1)=X(1)-X(2)
      IF (K1.EQ.2) GO TO 10
      A(1)=-(DY1+(Y(2)-Y(1))/B(1))/B(1)
      B(1)=0.0
10    J=1
      DO 20 I=1,N1
      K=I+1
      J=J+1
      C1=X(K)-X(I)
      C2=X(K+1)-X(K)
      C3=Y(K)-Y(I)
      C4=Y(K+1)-Y(K)
      C5=C3/C1-C4/C2
      C6=C1/C2
      C7=C1*C2
      B(J)=1.0/(C6*(C1-B(J-1)))
      A(J)=(C5/C2-C6*A(J-1))*B(J)
      J=J+1
      B(J)=1.0/((-C1-C2)/C7-C6*B(J-1))
20    A(J)=(-C5/C7-C6*A(J-1))*B(J)
      A(J+1)=(DY2-C4/C2+C2*A(J))/(C2*(B(J)-C2))
      IF (K2.EQ.2) A(J+1)=(DY2/2.+A(J))/(-2.*C2+B(J))
      J=2*(N-1)
30    J=J-1
      IF (J.LE.0) RETURN
      A(J)=A(J)-B(J)*A(J+1)
      GO TO 30
      END
```

```
      SUBROUTINE KURVE (A,X,Y,XP,YP,N,L)
      DIMENSION A(1),X(1),Y(1)
   10 DO 30 I=2,N
      IF ((XP-X(1))*(X(2)-X(1))) 40,40,20
   20 IF ((XP-X(1))*(XP-X(I))) 40,30,30
   30 CONTINUE
   40 K=2*I-3
      C1=XP-X(I-1)
      C2=X(I)-XP
      IF (L.EQ.8) C2=0.
      SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))
      YP=Y(I-1)+(SLOPE+A(K)*C2+A(K+1)*C1*C2)*C1
      RETURN
      END
```

```
      SUBROUTINE CONICF (P,Q,R,S,T,SG,X,Y,YP)
      ROOT=SG*SQRT((R*X+S)*X+T)
      Y=P*X+Q+ROOT
      IF ((SG.NE.0.).AND.(ROOT.NE.0.)) GO TO 10
      YP=P
      GO TO 20
10    YP=P+(R*X+S/2.0)/ROOT
20    RETURN
      END
```

```
      SUBROUTINE CUBIC (X,YI,Z,ZX,ZXP,A,B,C,F)
      DIMENSION ZX(1),ZXP(1)
      IF (Y.EQ.YI) GO TO 30
      Y=YI
      DZ1=ZX(8)-ZX(11)
      DZ2=ZX(9)-ZX(2)
      DZC=ZX(10)-ZX(11)
      DZU=ZX(1)-ZX(2)
      DZ21=ZX(9)-ZX(8)
      X1=(ZX(12)-ZX(13))*SQRT(1.0-(DZ1/DZC)**2)+ZX(13)
      X2=(ZX(5)-ZX(4))*SQRT(1.0-(DZ2/DZU)**2)+ZX(4)
      DX21=X2-X1
      IF (X1.GT.ZX(13)) GO TO 10
      Z1P=0.0
      X1Y=0.0
      Z1PY=0.0
      GO TO 20
10    Z1P=-X1/DZ1*(DZC/(ZX(12)-ZX(13)))**2
      X1P=1.0/Z1P
      X1Y=X1*((ZXP(12)-ZXP(13))/(ZX(12)-ZX(13)))+(ZXP(8)+((ZX(8)-ZX(10))
     1*ZXP(11)-DZ1*ZXP(10))/DZC)/Z1P+ZXP(13)
      X1PY=X1P*((ZXP(8)-ZXP(11))/DZ1-X1Y/X1+2.0*((ZXP(12)-ZXP(13))/(ZX(
     112)-ZX(13))-(ZXP(10)-ZXP(11))/DZC))
      Z1PY=-X1PY*Z1P/X1P
20    Z2P=-X2/DZ2*(DZU/(ZX(4)-ZX(5)))**2
      X2P=1.0/Z2P
      X2Y=X2*((ZXP(5)-ZXP(4))/(ZX(5)-ZX(4)))+(ZXP(9)+((ZX(9)-ZX(1))*ZXP(
     12)-DZ2*ZXP(1))/DZU)/Z2P+ZXP(4)
      X2PY=X2P*((ZXP(9)-ZXP(2))/DZ2-X2Y/X2+2.0*((ZXP(4)-ZXP(5))/(ZX(4)-
     1ZX(5))-(ZXP(1)-ZXP(2))/DZU))
      Z2PY=-X2PY*Z2P/X2P
      ZY21=ZXP(9)-ZXP(8)
      DX21Y=X2Y-X1Y
30    IF ((Z1P.LT.-4.0).OR.(Z2P.LT.-4.0).OR.(Z1P.GT.0.01)) GO TO 40
      CA=3.0*DZ21-(Z2P+2.0*Z1P)*DX21
      CB=-2.0*DZ21+(Z2P+Z1P)*DX21
      CAY=3.0*ZY21-(Z2PY+2.0*Z1PY)*DX21-(Z2P+2.0*Z1P)*DX21Y
      CBY=-2.0*ZY21+(Z2PY+Z1PY)*DX21+(Z2P+Z1P)*DX21Y
      DX1=X-X1
      DX2=X-X2
```

```
      T=DX1/DX21
      A=-Z1P-(2.0*CA+3.0*CB*T)*T/DX21
      B=-ZXP(8)-Z1PY*DX1+Z1P*X1Y-(CAY+CBY*T)*T*T-(A+Z1P)*(X2Y*DX1-X1Y*
     1DX2)/DX21
      C=1.0
      F=Z-ZX(8)-Z1P*DX1-(CA+CB*T)*T*T
      INDEX=1
      RETURN
   40 CA=3.0*DX21-(X2P+2.0*X1P)*DZ21
      CB=-2.0*DX21+(X2P+X1P)*DZ21
      CAY=3.0*DX21Y-(X2PY+2.0*X1PY)*DZ21-(X2P+2.0*X1P)*ZY21
      CBY=-2.0*DX21Y+(X2PY+X1PY)*DZ21+(X2P+X1P)*ZY21
      DZ1=Z-ZX(8)
      DZ2=Z-ZX(9)
      T=DZ1/DZ21
      A=1.0
      C=-X1P-(2.0*CA+3.0*CB*T)*T/DZ21
      B=-X1Y-X1PY*DZ1+X1P*ZXP(8)-(CAY+CBY*T)*T*T-(C+X1P)*(ZXP(9)*DZ1-ZXP
     1(8)*DZ2)/DZ21
      F=X-X1-X1P*DZ1-(CA+CB*T)*T*T
      INDEX=2
      RETURN
      END
```

```
      SUBROUTINE ELIPSE (X,Z,ZX,ZXP,IZK,IZM,IXK,IXM,A,B,C,F)
      DIMENSION ZX(1),ZXP(1)
      FF=Z-ZX(IZK)
      DX=X-ZX(IXK)
      DXK=ZX(IXM)-ZX(IXK)
      DZ=Z-ZX(IZM)
      DZK=ZX(IZK)-ZX(IZM)
      A=2.0*DX/DXK**2
      C=2.0*DZ/DZK**2
      B=C*(ZXP(IZM)*FF-ZXP(IZK)*DZ)/DZK+A*(ZXP(IXK)*(X-ZX(IXM))-ZXP(IXM)
     1*DX)/DXK
      F=(DZ/DZK)**2+(DX/DXK)**2-1.0
      RETURN
      END
```

```
      SUBROUTINE WINGF (ZX,ZXP,X,YI,Z,IXG,IXE,IZG,IZE,A,B,C,F)
      DIMENSION ZX(1),ZXP(1)
      IF ((Y.EQ.YI).AND.(IZG.EQ.IZGS)) GO TO 10
      Y=YI
      IZGS=IZG
      T1=ZX(IXG)-ZX(IXE)
      T2=ZX(IZG)-ZX(IZE)
      T3=T1/T2
      D=(T2*(ZXP(IXG)-ZXP(IXE))-T1*(ZXP(IZG)-ZXP(IZE)))/T2**2
   10 DZ=Z-ZX(IZE)
      F=(X-ZX(IXE))-T3*DZ
      A=1.0
      B=T3*ZXP(IZE)-DZ*D-ZXP(IXE)
      C=-T3
      RETURN
      END
```

```
      SUBROUTINE PCRCL (ZX,ZXP,X,YI,Z,IZE,IZG,IXG,IZU,IXU,IZM,IXM,IZ1,FX
     1,FY,FZ,F,X2,Z2)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT
      DIMENSION ZX(1),ZXP(1)
      IT=0
      IF ((ZX(IZ1).LT.ZX(2)).AND.(ZX(IZ1).GT.ZX(7))) IT=1
      IF ((Y.EQ.YI).AND.(ITT.EQ.IT).AND.(IZEP.EQ.IZE)) GO TO 50
      Y=YI
      ITT=IT
      IZEP=IZE
      XM4=(ZX(IZG)-ZX(IZE))/(ZX(IXG)-ZX(IXM))
      IF (IZG.EQ.16) SS=-SIGN(1.0,XM4)
      IF (IZG.EQ.17) SS=SIGN(1.0,XM4)
      ZUM=ZX(IZU)-ZX(IZM)
      XMU=ZX(IXM)-ZX(IXU)
      IF (IT.NE.1) GO TO 10
      X1=ZX(IXM)
      X1Y=ZXP(IXM)
      XM1=0.
      XM1Y=0.
      XM3P=0.
      XM3PY=0.
      GO TO 20
   10 Z1M=ZX(IZ1)-ZX(IZM)
      Z1MZUM=Z1M/ZUM
      T1=SQRT(1.0-Z1MZUM**2)
      X1=ZX(IXU)+XMU*T1
      X1U=X1-ZX(IXU)
      T2=2.0*Z1MZUM*((ZXP(IZ1)-ZXP(IZM))/ZUM-Z1M*(ZXP(IZU)-ZXP(IZM))/ZUM
     1**2)
      X1Y=-XMU**2/X1U*T2/2.0+T1*(ZXP(IXM)-ZXP(IXU))+ZXP(IXU)
      XMUZUM=XMU/ZUM
      T1=Z1M/X1U
      XM1=XMUZUM**2*T1
      T2=((ZXP(IXM)-ZXP(IXU))/ZUM-XMU*(ZXP(IZU)-ZXP(IZM))/ZUM**2)*T1
      T3=(ZXP(IZ1)-ZXP(IZM))/X1U-Z1M*(X1Y-ZXP(IXU))/X1U**2
      XM1Y=(2.0*T2+XMUZUM*T3)*XMUZUM
      XM3=-1.0/XM1
      XM3P=-1.0/XM3
      XM3Y=XM1Y/XM1**2
```

```
      XM3PY=XM3Y/XM3**2
   20 XM4P=-1.0/XM4
      XGE=ZX(IXG)-ZX(IXM)
      ZGE=ZX(IZG)-ZX(IZE)
      XM4Y=(ZXP(IZG)-ZXP(IZE))/XGE-(ZXP(IXG)-ZXP(IXM))*ZGE/XGE**2
      XM4PY=XM4Y/XM4**2
      DTH=(ATAN(XM3P)+SS*ATAN(XM4P))/2.0
      XM5=TAN(DTH)
      T1=(1.0/COS(DTH))**2
      T2=XM3PY/(1.0+XM3P**2)+XM4PY/(1.0+XM4P**2)
      XM5Y=T1*T2/2.0
      IF (IT.NE.1) GO TO 30
      XP=ZX(IXM)
      ZP=ZX(IZE)
      XPY=ZXP(IXM)
      ZPY=ZXP(IZE)
      GO TO 40
   30 ZE1=ZX(IZE)-ZX(IZ1)
      XM34=XM3-XM4
      T1=(XM4*ZX(IXM)-XM3*X1)
      XP=(ZE1-T1)/XM34
      T2=ZXP(IZE)-ZXP(IZ1)-XM4*ZXP(IXM)-XM4Y*ZX(IXM)+XM3*X1Y+XM3Y*X1
      XPY=(XM34*T2-(ZE1-T1)*(XM3Y-XM4Y))/XM34**2
      ZP=ZX(IZE)+XM4*(XP-ZX(IXM))
      ZPY=ZXP(IZE)+XM4*(XPY-ZXP(IXM))+(XP-ZX(IXM))*XM4Y
   40 XM15=XM1-XM5
      T1=(ZP-ZX(IZ1))-(XM5*XP-XM1*X1)
      T2=ZPY-ZXP(IZ1)-XM5*XPY-XM5Y*XP+XM1*X1Y+XM1Y*X1
      XO=T1/XM15
      XOY=(XM15*T2-T1*(XM1Y-XM5Y))/XM15**2
      ZO=ZX(IZ1)+XM1*(XO-X1)
      ZOY=ZXP(IZ1)+XM1*(XOY-X1Y)+XM1Y*(XO-X1)
      ZOZ1=ZO-ZX(IZ1)
      XOX1=XO-X1
      R2=ZOZ1**2+XOX1**2
      R=SQRT(R2)
      IF (IPRINT.GE.2) WRITE (IOU6,80) X,Y,Z,XO,ZO,R,XP,ZP,IT
   50 IF (X.LT.-1.) GO TO 60
      RY=(ZOZ1*(ZOY-ZXP(IZ1))+XOX1*(XOY-X1Y))/R
      XXO=X-XO
```

```
      ZZO=Z-ZO
      F=R2-ZZO**2-XXO**2
      FX=-2.0*XXO
      FY=2.0*(R*RY+XXO*XOY+ZZO*ZOY)
      FZ=-2.0*ZZO
      GO TO 70
60    CONTINUE
      D=SQRT((X1-XP)**2+(ZX(IZ1)-ZP)**2)
      ANG=ATAN(XM4)
      X2=XP+D*COS(ANG)
      Z2=ZP+XM4*(X2-XP)
70    RETURN
C
C
80    FORMAT (16HO*****PCRCL*****/(8G15.7))
      END
```

```
      SUBROUTINE PELIPS (ZX,ZXP,X,YI,Z,IZE,IZG,IXG,IXM,IZI,IXI,FX,FY,FZ,
     1F)
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT
      DIMENSION ZX(23),ZXP(23)
      IF ((YI.EQ.Y).AND.(IZES.EQ.IZE)) GO TU 10
      Y=YI
      IZES=IZE
      DZT=ZX(IZG)-ZX(IZE)
      DZB=ZX(IXG)-ZX(IXM)
      Z2P=DZT/DZB
      Z2PY=(DZB*(ZXP(IZG)-ZXP(IZE))-DZT*(ZXP(IXG)-ZXP(IXM)))/DZB**2
      C=ZX(IXI)-ZX(IXG)
      CY=ZXP(IXI)-ZXP(IXG)
      DZ=ZX(IZG)-ZX(IZI)
      DZY=ZXP(IZG)-ZXP(IZI)
      T1=C*(C+DZ/Z2P)
      T2=2.0*C+DZ/Z2P
      A=T1/T2
      T1Y=2.*C*CY+(Z2P*(C*DZY+CY*DZ)-C*DZ*Z2PY)/Z2P**2
      T2Y=2.0*CY+(Z2P*DZY-DZ*Z2PY)/Z2P**2
      AY=(T2*T1Y-T1*T2Y)/T2**2
      AMC=A-C
      B2=DZ**2-AMC*DZ*Z2P
      B2Y=2.0*DZ*DZY-AMC*(DZ*Z2PY+Z2P*DZY)-DZ*Z2P*(AY-CY)
      XO=ZX(IXI)-A
      XOY=ZXP(IXI)-AY
   10 IF (SQRT(B2).GT.1.5*ABS(ZX(IZG)-ZX(IZI))) GC TO 20
      IF (IPRINT.GT.1) WRITE (IOU6,30) B2,DZ,T1,T2,Z2P,IZE,YI,Z,ZX
      D=(X-XO)/A
      DY=(-A*XOY-(X-XO)*AY)/A**2
      ZMZ=Z-ZX(IZI)
      F=ZMZ**2+B2*(D**2-1.0)
      FY=2.0*(B2*D*DY-ZMZ*ZXP(IZI))+B2Y*(D**2-1.0)
      FX=2.0*D*B2/A
      FZ=2.0*ZMZ
      IF (IPRINT.GT.1) WRITE (IOU6,30) F,FX,FZ,A,B2,D,DY,XO,XOY
      RETURN
   20 ZSQ=SQRT(1.-(Z-ZX(IZI))**2/B2)
      F=X-XO-A*ZSQ
      FX=1.
```

```
      FZ=A/B2*(Z-ZX(IZI))/ZSQ
      FY=-XOY-AY*ZSQ-FZ*(ZXP(IZI)+.5*(Z-ZX(IZI))/B2*B2Y/B2)
      RETURN
C
C
30    FORMAT (17HO*****PELIPS**** /(8G15.7))
      END
```

```
      SUBROUTINE CONPIC (C,NC,XP,YP,DYP,IND)
      DIMENSION C(8,4)
      IND=0
      DO 10 I=1,NC
      IF ((XP.GE.C(1,I)).AND.(XP.LE.C(2,I))) GO TO 20
   10 CONTINUE
      IND=1
   20 CALL CONICF (C(3,I),C(4,I),C(5,I),C(6,I),C(7,I),C(8,I),XP,YP,DYP)
      RETURN
      END
```

```
      SUBROUTINE XSUBG (ZX,ZXP,R)
      DIMENSION ZX(1),ZXP(1)
      T1=ZX(21)-ZX(5)
      T2=ZX(16)-ZX(18)
      T3=ZX(14)-ZX(18)
      ZX(19)=ZX(21)-T1*T2/(R*T3)
      ZXP(19)=ZXP(21)-(T3*(T2*(ZXP(21)-ZXP(5))+T1*(ZXP(16)-ZXP(18)))-T1*
     1T2*(ZXP(14)-ZXP(18)))/(R*T3*T3)
      ZX(20)=ZX(19)
      ZXP(20)=ZXP(19)
      RETURN
      END
```

```
      SUBROUTINE AERO (NPC,SINA,COSA)
      COMMON /CP/ B(99,4,2),DRAG,ALIFT,TORQUE,YMT,ZMT,WINGA
      COMMON /IO/ IOU5,IOU6,IOU7,IPRINT,EPSLN,DP,DTHETA,DPSI,KP4,IOU3
      DO 10 I=1,2
      DO 10 J=2,NPC
      DO 10 K=1,2
      L=J+K-1
      M=(3-K)*I+(3-I)*K-3
      CPA=(B(L,4,M)+B(J+1,4,I)+B(J,4,3-I))/(6.0*WINGA)
      YA=(B(L,2,M)+B(J+1,2,I)+B(J,2,3-I))/3.0
      ZA=(B(L,3,M)+B(J+1,3,I)+B(J,3,3-I))/3.0
      IF (I.NE.K) CPA=-CPA
      DARY=CPA*((B(J+1,3,I)-B(L,3,M))*(B(J,1,3-I)-B(L,1,M))-(B(J+1,1,I)-
     1B(L,1,M))*(B(J,3,3-I)-B(L,3,M)))
      DARZ=CPA*((B(J+1,1,I)-B(L,1,M))*(B(J,2,3-I)-B(L,2,M))-(B(J+1,2,I)-
     1B(L,2,M))*(B(J,1,3-I)-B(L,1,M)))
      DRAG=DRAG+DARY*COSA+DARZ*SINA
      ALIFT=ALIFT+DARZ*COSA-DARY*SINA
   10 TORQUE=TORQUE+DARY*(ZA-ZMT)-DARZ*(YA-YMT)
      FY=DRAG*COSA-ALIFT*SINA
      FZ=DRAG*SINA+ALIFT*COSA
      WRITE (IOU6,20) DRAG,ALIFT,TORQUE,YMT,ZMT,FY,FZ
      RETURN
C
C
   20 FORMAT (1H0,T7,4HDRAG,T22,4HLIFT,T35,6HMOMENT,T47,13HPIVOT POINT Y
     1,T68,1HZ,T80,7HY-FORCE,T95,7HZ-FORCE/7G15.6)
      END
```

```
      SUBROUTINE SETCP (LNTAPE,IB,A,B,SINE,COSINE,DUMMY,GF,FMSTR,N2)
      DIMENSION A(99,8),B(99,4,2)
      REWIND LNTAPE
      READ (LNTAPE) ((A(L,M),M=1,8),L=2,N2)
      DO 10 I=2,N2
      B(I,1,IB)=COSINE*A(I,1)-SINE*A(I,2)
      B(I,2,IB)=SINE*A(I,1)+COSINE*A(I,2)
      B(I,3,IB)=A(I,3)
   10 B(I,4,IB)=2.0*(A(I,7)*DUMMY-1.0)/(GF*FMSTR*FMSTR)
      REWIND LNTAPE
      RETURN
      END
```

SECTION VII

FUNCTIONAL DESCRIPTION OF SUBROUTINES

BULK	-	Coordinates the three basic modes of computation: body point, field point, and shock point calculations. It establishes the position of a new point at each iteration and checks to see if the convergence criteria are satisfied.
PICK	-	Picks the appropriate data points from the last computed data surface for each of the three basic modes of computation.
BASEPT	-	Determines the location of the base point (the intersection of a line joining two data points with the backward facing Mach conoid) and the flow properties at this point.
COEFS	-	Computes the values of the coefficients of the compatibility relation in the generalized finite-difference form for each of the three basic computations.
COMPAT	-	Solves the nonlinear equation resulting from manipulation of the compatibility relations for the shock point computation.
GMTRY	-	Finds a new body point and the unit normal at that point.
FIGURE	-	Describes a slab-delta-wing type body through six conic-section curves and provides the unit normal at any specified body point.
TRANS	-	Transforms to and from a body-normal oriented coordinate system.
ROTATE	-	Rotates a two-dimensional vector through an angle.
SOL	-	Solves three simultaneous linear equations.
HARNES	-	Calculates the average between the locally computed shock normal and the shock normal determined from the shape of the shock surface and recomputes the local velocity behind the shock, using the average normal.
MIRROR	-	Sets the mirror point properties across the plane of symmetry.
NORM	-	Normalizes a vector.
DATOUT	-	Performs all output options.
TIDYUP	-	Rearranges the data points along the data rings.
REFORM	-	Rearranges the data rings on a data surface.
KURFIT	-	Provides the coefficients of a cubic spline for data interpolation.
KURVE	-	Performs data interpolation through a cubic spline.
ROTXY	-	Rotates the X-Y corrdinates.
CONICF	-	Computes Y and Y' from conic-section coefficients.
CUBIC	-	Computes the cubic contour near the fuselage-canopy juncture.

ELIPSE	-	Computes the elliptic contour of the fuselage.
WINGF	-	Computes the upper or lower wing contour.
PCRCL	-	Computes the upper blending circular contour.
PELIPS	-	Computes the partial ellipse contour near the wing leading edge.
CONPIC	-	Computes the upper limit point of the upper blending contour.
XSUBG	-	Determines the X abscissa of point G of Figure 15.
AERO	-	Computes the pressure coefficient and moments.
SETCP	-	Collects relevent data for AERO.
SECOND	-	Makes interpolation using the 2nd order Lagrange's formula.

PART 3: THE SURFACE FIT AND MINIMUM SEARCH PROGRAM

SECTION I

INTRODUCTION

Latin Squares are used to sample configurations for wave drag calculations. Corresponding to each cell of the Square, a configuration exists which can be represented by a set of body describing input data. Both 5 x 5 and 3 x 3 Latin Squares are used: the former is for Phase I study and the latter, Phase II study. Six variables $z_1, \ldots, z_6$ can be treated by a 5 x 5 Latin Square and four variables $z_1, \ldots, z_4$ by a 3 x 3 Latin Square. These reduced variables z_i are related to the physical variables x_i through the formula

$$x_i = \frac{x_{i_{max}} + x_{i_{min}}}{2} + \frac{z_i \left(x_{i_{max}} - x_{i_{min}}\right)}{n-2}$$

where n = 6 for a 5 x 5 Latin Square and 4 for a 3 x 3 Latin Square. The advantage of the reduced variables is that a program written in terms of them is valid for any physical variables with any ranges of variation.

When the wave drag coefficients C_{DW} have been calculated as input data, the Surface Fit and Minimum Search program is used to obtain the coefficients b_o, b_i, and b_{ij} of the wave drag equation*

$$C_{D_W} = b_o + \sum b_i z_i + \sum b_{ij} z_i z_j$$

through a least wquare procedure; in other words, the program is used to fit a hypersurface in an n dimensional space. When the type of constraints is specified, the program is used to yield a set of values for the z_i that corresponds to a minimum wave drag body subject to the given constraints.

---

*i and j range from 1 to n; however, as explained in Volume I, not every possible term is included.

SECTION II

GUIDE TO THE SEARCH PROCEDURE

In the search procedure, wave drag coefficients are computed using the wave drag equation at points covering the entire region bounded by the given ranges of variables in the n-dimensional space. The point which has the lease wave drag is taken to define the minimum wave drag configuration. The accuracy of locating the minimum is therefore dependent on the resolution of the numerical network. Accurate location of the minimum can be obtained efficiently by a search-by-steps technique where the search is first conducted using coarser grids to determine an approximate location, and then repeated in a smaller region centered around the location of the previously obtained minimum. This process can be repeated as many times as needed to obtain the desired accuracy. The major advantage is that the given constraints can be satisfied easily by rejecting points which violate the constraints.

1. SEARCH STEPS

The searching procedure is summarized as follows:

a. Select the ranges of search by specifying the lower and upper limits of each of the n number of reduced variables:*

$$\left(z_L\right)_j \,,\quad \left(z_u\right)_j \qquad j = 1,\ n \qquad \begin{cases} n = 4 \text{ for } 3 \times 3 \text{ Latin Square} \\ n = 6 \text{ for } 5 \times 5 \text{ Latin Square} \end{cases}$$

b. Select the resolution for each of the variables by specifying the numbers of grid points $N_j (j = 1,\ n)$ of that variable. The resolution is given by

$$\text{resolution} = \frac{\left(z_u\right)_j - \left(z_L\right)_j}{N_j - 1}$$

---

* $\left(z_L\right)_j \equiv O(J)$, $\left(z_u\right)_j \equiv E(J)$, and $N_j \equiv NP(J)$ in the input data described in Section III.

The total number of points for each search cycle is equal to the product of N_j, i.e.,

$$\text{total points} = \prod_{j=1}^{n} N_j$$

For example, if 10 points are specified for all variables, the total number of points for n = 6 is 10^6. To keep the searching time within practical limit, N_j must therefore be selected carefully so that the total number of searching points is within reasonable limits.* Using the data of a. and b., the program computes the n tables of variables:

$$\left(z_i\right)_j \quad \left\{ \begin{matrix} i = 1, N_j \\ \left(z_1\right)_j = \left(z_L\right)_j ; \left(z_N\right)_j = \left(z_u\right)_j \end{matrix} \right\} j = 1, n$$

so that all configurations defined by every possible combination of the n variables,

$$\left(z_i\right)_1, \left(z_i\right)_2, \ldots\ldots\ldots, \left(z_i\right)_n$$

can be checked for the given constraints (e.g., the volume). The configuration which has the minimum wave drag coefficient computed by the wave drag equation is considered the minimum wave drag configuration.

c. If the search is to be repeated, select the new ranges of the n variables by redefining the lower and upper limits as,

$$\left(z_L\right)_j = z_j' - \Delta_j, \left(z_u\right)_j = z_j' + \Delta_j$$

where z_j' are the variables for the minimum wave drag configuration determined previously, and Δ_j are arbitrary increments of z_j selected to define a smaller region of search. In situations where $\left(z_L\right)_j$ or $\left(z_u\right)_j$ defined by

*See Section IV for time estimates.

above relations are beyond the ranges of consideration, they are then set equal to the minimum or maximum of the range.

d. Repeat the procedure from step b. until the desired accuracy is obtained.

2. TYPES OF CONSTRAINTS

Four types of geometric constraints are included in order to satisfy most design needs. These are:

a. One or more of the six geometric variables take assigned values within the ranges of variation. For instance, the length can be kept the same as that of the baseline or the width can take 90% of the baseline width.

b. At one or more fuselage stations the configuration cross sections contain geometric curves prescribed by tabulated data of width X versus elevation Z. This type of constraint is useful for the placement of equipment. In the case of the radome constraint, only the radome radius and the elevation of the center need to be specified.

c. At one or more fuselage stations the configuration cross sections satisfy minimum area requirements. The minimum area can be either a given area or a certain percentage of the baseline cross-sectional area.

d. The configuration satisfies a minimum volume requirement between two given stations. A given volume or a certain percentage of the baseline volume can be imposed.

SECTION III

INPUT/OUTPUT DESCRIPTION

Two different sizes of Latin Squares are used in Phase I and Phase II; the input and output of these two phases are described in this section.

1. INPUT DESCRIPTION

The input data cards for Phase I program LATIN1 and Phase II program LATIN2 are shown in Figures 22 and 23. The cards are described below:

Phase I Program LATIN1 Input

Card No	Variable	Format	Description
1	SYM(1) ⋮ SYM(6)	3 x A2 ⋮ 3 x 2A	Alphanumeric symbols to identify reduced variables $z_1, \ldots, z_6$ in print out
2	IOR	15	Number of terms in the Wave Drag equation
	IPRINT	15	Print out control
			IF IPRINT = 0, Minimum Print; = 1, detail print about surface fit
3	KIND	15	Constraint Control
			If KIND = 0, no constraint; or constraints imposed on one or more of the variables; = 1, local area-ratio constraint; = 2, local tabulated X-vrs-Z constraint; = 3, randome constraint; = 4, volume constraint

- Use card 4_1 if KIND = 1 or 2

Card								
①	S_1	S_2	S_3	S_4	S_5	S_6	S_i = SYM(I)	
②	IOR	IPRINT						
③	KIND							
	● Use 4_1 if KIND = 1 or 2							
4_1	FS	RATIO						
	● Add 4_2 and 5_2 if KIND = 2							
4_2	NPFS							
5_2	ZFS(1)	XFS(1)	ZFS(2)	XFS(2)	. . . . .	. . . . .	ZFS(NPFS)	XFS(NPFS)
	● Use 4_3 if KIND = 3							
4_3	FS	RC	RD					
	● Use 4_4 and 5_4 if KIND = 4							
4_4	FS1	FS2	ANY	NFSI				
5_4	FS(2)	FS(3)	. . . .	. . . .	FS(NFSI - 1)			
⑥	(Blank Card)							

Card								
⑦	1	CD(1)						
	:	:						
	;	:						
	25	CD(25)						
⑧	26	CD(26)	Z_1(26)	Z_2(26)	Z_3(26)	Z_4(26)	Z_5(26)	Z_6(26)
	:	:	.	:	:	:	:	:
	:	:	.	:	:	:	:	:
	J	CD(J)	Z_1(J)	Z_2(J)	Z_3(J)	Z_4(J)	Z_5(J)	Z_6(J)
⑨	(Blank Card)							

FIGURE 22. INPUT DATA FOR SURFACE FIT AND MINIMUM SEARCH PROGRAM USING 5 x 5 LATIN SQUARE

	● Use 10 through 12 only if KIND = 4					
⑩	1	Volume(1)				
		:				
⑪		:				
	J	Volume(J)				
⑫	(Blank Card)					
⑬	NP(1)	NP(2)	NP(3)	NP(4)	NP(5)	NP(6)
⑭	O(1)	O(2)	O(3)	O(4)	O(5)	O(6)
⑮	E(1)	E(2)	E(3)	E(4)	E(5)	E(6)

FIGURE 22. (Continued)

Card	Columns 1–10	Columns 11–20	Columns 21–30	Columns 31–40	Columns 41–50	Columns 51–60
①	S_1 (col. 5)	S_2 (col. 12)	S_3 (col. 18)	S_4 (col. 24)		S_i = SYM(I)
②	IOR, IPRINT					
③	KINDT, VOLBIN					
④	1 CD(1)					
	: :					
	: :					
	9 CD(9)					
⑤	10 CD(10)	Z_1(10)	Z_2(10)	Z_3(10)	Z_4(10)	
	: :	:	:	:	:	
	: :	:	:	:	:	
	J CD(J)	Z_1(J)	Z_2(J)	Z_3(J)	Z_4(J)	
⑥	(Blank Card)					
	● Use 7 through 9 only if KIND = 1					
⑦	1 Volume(1)					
	: :					
⑧	: :					
	J Volume(J)					
⑨	(Blank Card)					
⑩	NP(1) NP(2) NP(3) NP(4)					
⑪	O(1)	O(2)	O(3)	O(4)		
⑫	E(1)	E(2)	E(3)	E(4)		

FIGURE 23. INPUT DATA FOR SURFACE FIT AND MINIMUM SEARCH PROGRAM USING 3 x 3 LATIN SQUARE

Card No	Variable	Format	Description
4_1	FS	F9. 2	Fuselage station at which cross-sectional area constraint is imposed
	RATIO	F9. 2	Ratio of desired area to baseline cross-sectional area at this F. S. if $0 < \text{RATIO} \leq 2$ Desired cross-sectional area at this F. S. if RATIO > 2 Not used if KIND = 2; may be left blank

• Add cards 4_2 and 5_2 if KIND = 2

Card No	Variable	Format	Description
4_2	NPFS	I5	Number of points in the table
5_2	ZFS(1)	F9. 2	Elevation Z(1) of the first point
	XFS(1)	F9. 2	Width X(1) of the first point
	⋮	⋮	
	ZFS(NPFS	F9. 2	Elevation Z(NPFS) of the NPFSth point
	XFS(NPFS)	F9. 2	Width X(NPFS) of the NPFSth point

• Use card 4_3 if KIND = 3

Card No	Variable	Format	Description
4_3	FS	F9. 2	Fuselage station at which the radome constraint is imposed
	FC	F9. 2	Z value of radome center
	RD	F9. 2	Radius of Radome

• Use cards 4_4 and 5_4 if KIND = 4

Card No	Variable	Format	Description
4_4	FS1	F10. 2	Fuselage stations between which the volume constraints is imposed. Note that if FS1 < 50 the complete nose is included.
	FS2	F10. 2	
	ANY	F10. 2	Ratio of desired volume to baseline volume if $0 < \text{ANY} \leq 2$ Desired volume if ANY > 2
	NFSI	I5	Number of specified fuselage stations at which cross-sectional areas are calculated for volume integration
5_4	FS(2) ⋮ FS(NFSI-1)	F9. 2 ⋮ F9. 2	Fuselage Stations at which cross-sectional areas are calculated for volume integration
6	Blank Card		Must be present to terminate the input string

Card No	Variable	Format	Description
7	J	I2	Cell number of the Latin Square
	ANY	F10.0	Wave Drag Coefficient for Jth cell

- There follow 24 cards like card 7

Card No	Variable	Format	Description
8	J	I2	Identification number (starting from 26) for additional input of wave drag coefficient
	ANY	F10.0	Wave drag coefficient for Jth addition
	V(1)	F10.0	Value of reduced variable z_1 for Jth addition
	⋮	⋮	
	V(6)	F10.0	Value of reduced variable z_6 for Jth addition

- There follow additional cards like card 8

Card No	Variable	Format	Description
9	Blank card		Must be present to terminate the input string.

- If KIND $\neq$ 4, go to card 13

Card No	Variable	Format	Description
10	J	I2	Cell number of the Latin Square
	ANY	F10.0	Nose volume up to F.S. 50 for Jth cell*

- There follow 24 cards like card 10

Card No	Variable	Format	Description
11	J	I2	Identification number for additional nose volume input
	ANY	F10.0	Nose volume for Jth addition

- There follow additional cards like card 11

Card No	Variable	Format	Description
12	Blank card		Must be present to terminate the input string
13	NP(1)	I5	Number of evenly spaced grid points for reduced variable z_1 for the search process
	⋮	⋮	
	NP(6)	I5	Number of evenly spaced grid points for reduced variable z_6 for the search process
14	O(1)	F9.2	Lower limit of the search interval for z_1
	⋮	⋮	
	O(6)	F9.2	Lower limit of the search interval for z_6

*See Section IV for volume calculations.

Card No	Variable	Format	Description
15	E(1)	F9.2	Upper limit of the search interval for z_1
	⋮	⋮	
	E(6)	F9.2	Upper limit of the search interval for z_6

Phase II Program LATIN2 Input

Card No	Variable	Format	Description
1	SYM(1)	3 x A2	Alphanumeric symbols to identify reduced variables $z_1, \ldots, z_4$ in print out
	⋮	⋮	
	SYM(4)	3 x A2	
2	IOR	I5	Number of terms in the wave drag equation
	IPRINT	I5	Print out control If IPRINT = 0, minimum print = 1, detail print about surface fit
3	KINDT	I5	Constraint control If KINDT = 0, no constraints; or constraints imposed on one or more of the variables = 1, volume constraint
	VOLBIN	F10.0	Volume increment desired (i.e., the desired volume minus the baseline volume)*
4	J	I2	Cell number of the Latin Square
	ANY	F10.0	Wave drag coefficient for Jth cell

• There follow 8 cards like card 4

Card No	Variable	Format	Description
5	J	I2	Identification number (starting from 10) for additional input of wave drag coefficients
	ANY	F10.0	Wave drag coefficient for Jth addition

*See Section IV for volume calculations.

Card No	Variable	Format	Description
5	V(1)	F10.0	Value of reduced variable z_1 for Jth addition
	⋮	⋮	
	V(4)	F10.0	Value of reduced variable z_4 for Jth addition

- There follow additional cards like card 5

6	Blank Card		Must be present to terminate the input string.

- If KINDT = 0, go to card 10

7	J	I2	Cell number of the Latin Square
	ANY	F10.0	Volume increment for Jth cell (i. e., volume of Jth configuration minus the baseline volume)

- There follow 8 cards like card 7

8	J	I2	Identification number for additional volume increment input
	ANY	F10.0	Volume increment for Jth addition

- There follow additional cards like card 8

9	Blank card		Must be present to terminate the input string
10	NP(1)	I5	Number of evenly spaced grid points for reduced variable z_1 for the search process
	⋮	⋮	
	NP(4)	I5	Number of evenly spaced grid points for reduced variable z_4 for the search process
11	O(1)	F9.2	Lower limit of the search interval for z_1
	⋮	⋮	
	O(4)	F9.2	Lower limit of the search interval for z_4
12	E(1)	F9.2	Upper limit of the search interval for z_1
	⋮	⋮	
	E(4)	F9.2	Upper limit of the search interval for z_4

2. OUTPUT DESCRIPTION

Sample printouts of LATIN1 (5 x 5 Latin Square) and LATIN2 (3 x 3 Latin Square) are shown in Figures 24 through 27. Figure 24 shows the results of the drag coefficient surface fit for the 5 x 5 Latin Square. The input data shown in Figure 24 includes the drag coefficient for the basic 25 Latin Square cells plus 7 additional configurations including the baseline (No. 26). With the 7 additional drag coefficients, 6 additional terms were added to the wave drag equation (see Appendix A of Volume 1). Figure 25 shows the 5 x 5 Latin Square surface fit of the nose volume. The printout is similar to that of the drag coefficient. Figure 26 shows the results of a minimum wave drag configuration search with volume constraint. Here the search was made keeping the variable z_1 at a constant value of 0, while the remaining 5 variables $z_2, \ldots, z_5$ were varied through 5 discreet points between -2.0 and 2.0. That is, the resolution of each of the 5 searching variables in this search cycle is equal to 1.0. The minimum wave drag configuration, within the given ranges of variables and resolution is printed along with its cross-sectional areas and volume. The number of configurations which do not satisfy the volume constraint is indicated by the number of points rejected. Figure 27 shows a sample output of the 3 x 3 Latin Square surface fit and the minimum value search results with volume constraint. The printout is similar to that of the 5 x 5 Latin Square results. Here, however, the fitted volume is the complete fuselage canopy and wing, and the numerical values shown are the differences from the baseline.

```
                    LATIN SQUARE MINIMUM VALUE SEARCH PROGRAM
VARIABLES ARE LISTED AS. . . L     B     F     H     A*    A
ORDER OF FIT. . . .23
CONSTRAINT = 4-VOLUME

ADDITIONAL TERMS ARE. . . .
(18)    L     L
(19)    A     A
(20)    L     H
(21)    B     A
(22)    L     B
(23)    H     A

INPUT DATA.... *WAVE DRAG*
          1   8.722590
          2   5.970170
          3   5.724920
          4   5.884690
          5   5.952560
          6   7.625800
          7   7.254040
          8   7.105100
          9   5.439380
         10   5.760620
         11   6.526840
         12   6.673050
         13   6.591540
         14   6.861430
         15   7.610590
         16   8.217730
         17   8.688300
         18   6.438170
         19   6.618100
         20   5.939600
         21   7.630460
         22   8.222750
         23   8.625220
         24   7.248460
         25   5.876170
         26   6.694370
         27   6.572600
         28   6.095690
         29   5.955290
         30   5.863430
         31   6.814490
         32   6.118840
                    COEFFICIENTS OF WAVE DRAG EQUATION
( 1)  6.589280       ( 2) -.3965690       ( 3)  .1952752       ( 4)  .2064617E-02  ( 5)  .4552590
( 6) -.2050504E-01   ( 7)  .2688061       ( 8)  .3790653E-01   ( 9) -.4525385E-03  (10)  .4179457E-01
(11)  .2906677E-01   (12) -.2480445E-01   (13)  .5690592E-01   (14)  .1303170E-01  (15) -.1453514E-01
(16) -.2082480E-01   (17) -.4708812E-02   (18)  .4236248E-01   (19)  .1710669E-01  (20) -.4901643E-01
(21)  .2862795E-01   (22)  .1614830E-01   (23)  .1834393E-02   (
STD DEV=  .71811E-01
                    FITTED WAVE DRAG
( 1)  8.657905       ( 2)  5.936778       ( 3)  5.756353       ( 4)  5.968195      ( 5)  5.927520
( 6)  7.716832       ( 7)  7.130031       ( 8)  7.199410       ( 9)  5.445890      (10)  5.702036
(11)  6.609727       (12)  6.476629       (13)  6.589280       (14)  6.947679      (15)  7.551826
(16)  8.263872       (17)  8.683826       (18)  6.355805       (19)  6.736829      (20)  5.841928
(21)  7.645751       (22)  8.136829       (23)  8.601087       (24)  7.299810      (25)  5.939397
(26)  6.782620       (27)  6.607058       (28)  6.056381       (29)  5.948000      (30)  5.862854
(31)  6.801438       (32)  6.143416       (

                    ERROR OF FIT
( 1) -.6468454E-01   ( 2) -.3339190E-01   ( 3)  .3143257E-01   ( 4)  .8350464E-01  ( 5) -.2503977E-01
( 6)  .9103227E-01   ( 7) -.1240094       ( 8)  .9430971E-01   ( 9)  .6510390E-02  (10) -.5858410E-01
(11)  .8288685E-01   (12) -.1964207       (13) -.2259986E-02   (14)  .8624890E-01  (15) -.5876402E-01
(16)  .4614162E-01   (17) -.4473969E-02   (18) -.8236504E-01   (19)  .1187293      (20) -.9767153E-01
(21)  .1529119E-01   (22) -.8592102E-01   (23) -.2413316E-01   (24)  .5134994E-01  (25)  .6322689E-01
(26)  .8824958E-01   (27)  .3445787E-01   (28) -.3930931E-01   (29) -.7290010E-02  (30) -.5763780E-03
(31) -.1305239E-01   (32)  .2457550E-01   (
```

FIGURE 24. SURFACE FIT OF WAVE DRAG COEFFICIENTS (5 x 5 LATIN SQUARE)

```
INPUT DATA.... *VOLUME*
         1   27969.00
         2   24190.00
         3   22492.00
         4   27633.00
         5   31198.00
         6   25953.00
         7   26918.00
         8   31499.00
         9   25273.00
        10   30202.00
        11   22460.00
        12   26212.00
        13   29430.00
        14   34228.00
        15   37811.00
        16   28259.00
        17   33819.00
        18   29035.00
        19   32614.00
        20   30919.00
        21   26603.00
        22   31917.00
        23   35621.00
        24   36351.00
        25   31103.00
        26   25076.00
        27   25055.00
        28   23565.00
        29   21192.00
        30   20452.00
        31   27049.00
        32   27194.00
( 1)   29697.09     ( 2)   1468.806     ( 3)   854.0985     ( 4)  -39.72363     ( 5)   1778.234
( 6)  -189.2461     ( 7)   1423.439     ( 8)  -195.7796     ( 9)   56.59233     (10)   202.7722
(11)  -184.8959     (12)   111.8339     (13)  -189.0630     (14)   260.7751     (15)   108.9358
(16)   230.2987     (17)   325.6316     (18)  -54.28646     (19)   114.5688     (20)  -288.7734
(21)   372.5243     (22)  -251.2878     (23)   320.3092     (
STD DEV=  212.15
( 1)   28199.48     ( 2)   23933.67     ( 3)   22326.73     ( 4)   27831.99     ( 5)   31019.06
( 6)   25911.69     ( 7)   26861.34     ( 8)   31465.42     ( 9)   25708.52     (10)   30171.99
(11)   22483.77     (12)   25957.68     (13)   29697.09     (14)   33702.02     (15)   37972.44
(16)   28234.93     (17)   33408.26     (18)   29127.16     (19)   32664.66     (20)   30918.35
(21)   26743.74     (22)   31698.24     (23)   35829.78     (24)   36640.07     (25)   30867.88
(26)   25439.63     (27)   25164.94     (28)   23504.00     (29)   21128.77     (30)   20444.52
(31)   26999.85     (32)   27234.34     (

                          ERROR OF FIT
( 1)   230.4782     ( 2)  -256.3349     ( 3)  -165.2720     ( 4)   198.9875     ( 5)  -178.9408
( 6)  -41.31367     ( 7)  -56.65509     ( 8)  -33.57555     ( 9)   435.5181     (10)  -30.01195
(11)   23.77196     (12)  -254.3200     (13)   267.0949     (14)  -525.9835     (15)   161.4449
(16)  -24.07026     (17)  -410.7430     (18)   92.16195     (19)   50.65513     (20)  -.6518263
(21)   140.7384     (22)  -218.7605     (23)   208.7771     (24)   289.0659     (25)  -235.1225
(26)   363.6349     (27)   109.9404     (28)  -61.00450     (29)  -63.22808     (30)  -7.476213
(31)  -49.14677     (32)   40.34173     (
```

FIGURE 25. SURFACE FIT OF NOSE VOLUME (5 x 5 LATIN SQUARE)

```
INPUT NO. OF PTS.
  1    5    5    5    5    5
INPUT LOWER LIMITS
 0.      -2.0      -2.0      -2.0      -2.0      -2.0
INPUT UPPER LIMITS
 2.0      2.0       2.0       2.0       2.0       2.0
     1061 POINTS REJECTED
     2064 POINTS ACCEPTED
          AVERAGE=  7.2638

       MINIMUM DRAG CONFIGURATION

    CD               L                B                F                H                A*               A
 6.114940       0.                -1.000000        2.000000        -1.000000        1.000000         0.

KIND.FS.ABASE.AMIN     4  50.000      745.94      705.32

KIND.FS.ABASE.AMIN     4  90.000      1003.5      1005.7

KIND.FS.ABASE.AMIN     4  150.00      1438.0      1437.7

KIND.FS.ABASE.AMIN     4  190.00      1641.8      1650.0

KIND.FS.ABASE.AMIN     4  200.00      1685.3      1697.6

KIND.FS.ABASE.AMIN     4  205.00      1706.5      1720.9

KIND.FS.ABASE.AMIN     4  210.00      1747.9      1765.1

KIND.FS.ABASE.AMIN     4  215.00      1782.1      1801.7

KIND.FS.ABASE.AMIN     4  220.00      1775.4      1796.0

KIND.FS.ABASE.AMIN     4  225.00      1768.6      1790.1

KIND.FS.ABASE.AMIN     4 -0.          272.37      256.48

KIND.FS.ABASE.AMIN     4  230.00      1761.8      1784.2

VOLUME - BASELINE.MIN. CONF..NOSE...  264920.7      266021.9      26088.12
```

FIGURE 26. MINIMUM VALUE RESULTS WITH VOLUME CONSTRAINT (5 x 5 LATIN SQUARE)

```
INPUT ORDER OF FIT - 8 THRU 14
 IOR,IPRINT          10          1

VOLUME CONSTRAINT, VOLUME     116486.8     GREATER THAN BASELINE

ADDITIONAL TERMS ARE. . . .
(10)    H H

INPUT DATA.... *WAVE DRAG*
         1   .1046020E-01
         4   .9455500E-02
         7   .1166290E-01
         2   .1165250E-01
         5   .1040130E-01
         8   .1142970E-01
         3   .1078680E-01
         6   .1283200E-01
         9   .1143090E-01
        10   .9793590E-02
                  COEFFICIENTS OF WAVE DRAG EQUATION
( 1)   .1040130E-01  ( 2)   .2706667E-03  ( 3)   .5785167E-03  ( 4)   .3638333E-03  ( 5)   .7459000E-03
( 6)   .2598000E-03  ( 7)   .5391000E-03  ( 8)  -.1181679E-03  ( 9)   .2227321E-03  (10)   .6171788E-04
STD DEV=  .69843E-15
                  FITTED WAVE DRAG
( 1)   .1046020E-01  ( 2)   .1165250E-01  ( 3)   .1078680E-01  ( 4)   .9455500E-02  ( 5)   .1040130E-01
( 6)   .1283200E-01  ( 7)   .1166290E-01  ( 8)   .1142970E-01  ( 9)   .1143090E-01  (10)   .9793590E-02

                  ERROR OF FIT
( 1)  -.2775558E-15  ( 2)  -.6661338E-15  ( 3)  -.2775558E-15  ( 4)  -.6661338E-15  ( 5)  -.1609823E-14
( 6)  -.6106227E-15  ( 7)  -.3885781E-15  ( 8)  -.6661338E-15  ( 9)  -.1665335E-15  (10)  -.4996004E-15
****************************************

INPUT DATA.... *VOLUME*
         1  -70402.30
         4  -48181.60
         7   32094.00
         2   1265.500
         5   36291.10
         8   83446.10
         3   47920.10
         6   137459.1
         9   154575.0
        10   0.
( 1)   36291.10      ( 2)   48555.30      ( 3)   71074.02      ( 4)   7140.667      ( 5)   14605.67
( 6)   4258.567      ( 7)   2179.267      ( 8)   8759.461      ( 9)   8386.327      (10)  -6849.644
STD DEV=  .32892E-08
( 1)  -70402.30      ( 2)   1265.500      ( 3)   47920.10      ( 4)  -48181.60      ( 5)   36291.10
( 6)   137459.1      ( 7)   32094.00      ( 8)   83446.10      ( 9)   154575.0      (10)  -.7275958E-09

                  ERROR OF FIT
( 1)   .2793968E-08  ( 2)   0.            ( 3)  -.2328306E-08  ( 4)   .2095476E-08  ( 5)  -.1396984E-08
( 6)  -.6519258E-08  ( 7)  -.6984919E-09  ( 8)  -.3725290E-08  ( 9)  -.5587935E-08  (10)  -.7275958E-09
****************************************
****************************************
INPUT NO. OF PTS.
  21   21   21   21
INPUT LOWER LIMITS
-1.0      -1.0      -1.0      -1.0
INPUT UPPER LIMITS
 1.0       1.0       1.0       1.0
   179282 POINTS REJECTED
    15199 POINTS ACCEPTED
          AVERAGE=  .11728E-01

        MINIMUM DRAG CONFIGURATION

    CD              H              A               R1              R2
 .1074085E-01    .6000000      -.9000000      -.8000000       .9000000

 116486.8        120002.2
```

FIGURE 27. SURFACE FIT AND MINIMUM VALUE SEARCH RESULTS WITH VOLUME CONSTRAINT (3 x 3 LATIN SQUARE)

SECTION IV

OPERATIONAL ASPECTS

1. CORE AND TIME REQUIREMENTS

Both the LATIN1 and LATIN2 can be run on $60{,}000_8$ words of memory. The computer time required for the search is dependent on the total number of search cases. As a guide it takes about 5-10 seconds for the LATIN1 program and 0.05-0.10 seconds for the Latin2 program to compute 1000 search cases. Additional 5 seconds should be allowed for compilation.

2. ERROR MESSAGES

If the number of input data is less than the requested order of Latin Square fit (number of terms in the wave drag equation) the error message,

INSUFFICIENT NO. OF INPUT DATA

will be printed and the job will be terminated.

3. VOLUME DATA PREPARATION

If the volume constraint option is used, the two minimum value search programs require the input of configuration volumes as well as wave drag coefficients. For the LATIN1 program, the input volume is that of the fuselage nose ahead of fuselage station 50, while for the LATIN2 program, the input volume is that of the complete configuration which comprises the fuselage, canopy and wings. Two auxiliary programs, which use the same body discription data for the 3DMoC program, can be used to calculate the input volume data. The listings of the two volume calculation programs, one for LATIN1 and the other for LATIN2 programs, are presented below.

One input-data card is required for each of the volume calculation programs. The data required are

KASE, Y, Y2   FORMAT(I5, 2F10.2)

where KASE = an input integer to identify the body configuration (e.g., cell number)

Y1, Y2 = the fuselage stations between which the volume is to be calculated.

The values of Y1 and Y2 used in these two programs are

	LATIN1	LATIN2
Y1	0	0
Y2	50	430

```
C***FUSELAGE VOLUME CALCULATION PROCEDURE
      DIMENSION ZX(13),P(13),Q(13),R(13),S(13),T(13),SG(13),ZXP(13)
      DIMENSION AREA(101),YS(101)
      REWIND 2
      NY=101
      M78=10
      YMAXL=-100.
      YMAX=-100.
   10 WRITE(6,600)
  600 FORMAT(' INPUT KASE,Y1,Y2')
      READ(5,*) KASE,Y1,Y2
      XN=NY-1
      DELTAY=(Y2-Y1)/XN
      DO 50 LL=1,NY
      XLL=LL-1
      Y=Y1+DELTAY*XLL
       YS(LL)=Y
      DO 2 I=1,50
      IF((Y.LE.YMAX).AND.(Y.GE.YMAXL)) GO TO 3
      IF(Y.LT.YMAXL) REWIND 2
      YMAXL=YMAX
      READ(2,200) YMAX,N
      M=7
      DO 2 K=1,N
      READ(2,201) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
      IF(L.GT.7) M=13
      IF((L.EQ.9).OR.(L.EQ.8)) M78=8
    2 CONTINUE
  200 FORMAT(F10.5,I5)
  201 FORMAT(I2,5E15.8,F2.0)
    3 DO 4 I=1,7
      CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
    4 CONTINUE
      IF(M.EQ.7) GO TO 6
      DO 5 I=M78,13
    5 CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
      IF(M78.EQ.10) SG(8) = 99.
    6 CALL PLOT(Y,ZX,ZXP,M,SG,AREA(LL))
      MODE=1
   50 CONTINUE
      CALL SIGMA2(NY,0,YS,AREA,VOL)
      WRITE(6,700) KASE,VOL
  700 FORMAT(I5,F15.6)
```

```
      READ(5,500) DUMMY
  500 FORMAT(F20.0)
      GO TO 10
      END
```

```
      SUBROUTINE CONICF(P,Q,R,S,T,SG,X,Y,YP)
      ROOT=SG*SQRT((R*X+S)*X+T)
      Y=P*X+Q+ROOT
      IF(SG.NE.0.) GO TO 10
      YP=P
      GO TO 20
   10 YP=P+(R*X+S/2.0)/ROOT
   20 RETURN
      END
```

```
      SUBROUTINE PLOT(Y,ZX,ZXP,M,SG,AREA)
      COMMON PRINT,NMAX
      DIMENSION XP(200),ZP(200)
      DIMENSION ZX(1),ZXP(1),SG(1)
      N=200
      XN=N-1
      NM1=N-1
      IF(M.EQ.7) Z1=ZX(1)
      IF(M.EQ.13) Z1=ZX(10)
      Z2=ZX(3)
      XP(1)=0.
      XP(N)=0.
      ZP(1)=Z1
      ZP(N)=Z2
      DZ=(Z1-Z2)/XN
      DO 1000 I=2,NM1
      XI=I-1
      Z=Z1-DZ*XI
      ZP(I)=-Z
      IF((M.EQ.7).OR.(SG(8).EQ.99.)) GO TO 10
      IF(Z.GT.ZX(8)) GO TO 100
      IF(Z.GT.ZX(9)) GO TO 200
   10 IF(Z.GT.ZX(2)) GO TO 300
      IF(Z.GT.ZX(3)) GO TO 400
C***LOWER FLAT
      INDEX=5
      X=ZX(6)
      GO TO 500
C***CANOPY
  100 INDEX=1
      X=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),Z)+ZX(13)
      GO TO 500
C***CUBIC
  200 INDEX=2
      IF(SG(8).NE.99.) GO TO 15
      IF(Z.GE.ZX(1)) GO TO 300
      GO TO 400
   15 X1=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),ZX(8))+ZX(13)
      X2=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),ZX(9))+ZX(4)
      Z1P=XELIPP(ZX(11),ZX(10),X1,ZX(12),ZX(8))
      Z2P=XELIPP(ZX(2),ZX(1),X2,ZX(5)-ZX(4),ZX(9))
      DZ1=Z-ZX(8)
      DZ21=ZX(9)-ZX(8)
```

```
      A=3.*DZ21-(Z2P+2.*Z1P)*(X2-X1)
      B=-2.*DZ21+(Z2P+Z1P)*(X2-X1)
      IF((Z1P.LT.-4.).OR.(Z2P.LT.-4.).OR.(Z1P.GE.0.)) GO TO 22
      GO TO 18
   22 TX=1
      IF(Z.EQ.ZX(8))  X=X1
      IF(Z.EQ.ZX(9))  X=X2
      IF((Z.EQ.ZX(8)).OR.(Z.EQ.ZX(9))) GO TO 500
      XI=X1
      DO 19 L=1,2
      DO 17 J=1,6
      DXR=(XI-X1)/(X2-X1)
      ZI=ZX(8)+Z1P*(XI-X1)+A*DXR**2+B*DXR**3
      ZIP=Z1P+(2.0*A*DXR+3.0*B*DXR**2)/(X2-X1)
      IF(ZIP.NE.0.) GO TO 16
      X=XI+(X2-X1)/100.
      GO TO 17
   16 X=XI-(ZI-Z)/ZIP
      IF(ABS(X-XI).LE.1.E-5) GO TO 500
   17 XI=X
      XI=X2
   19 CONTINUE
      GO TO 500
   18 DX21=X2-X1
      X1P=1.0/Z1P
      X2P=1.0/Z2P
      A=3.0*DX21-(X2P+2.0*X1P)*DZ21
      B=-2.0*DX21+(X2P+X1P)*DZ21
      DZR=(Z-ZX(8))/DZ21
      X=X1+X1P*DZ1+A*DZR**2+B*DZR**3
      GO TO 500
C***UPPER ELLIPSE
  300 INDEX=3
      X=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),Z) + ZX(4)
      GO TO 500
C***LOWER CONIC
  400 INDEX=4
      CALL CONICS(ZX,ZXP,6,3,5,2,7,Z,X)
  500 XP(I)=X
 1000 CONTINUE
      CALL SIGMA2(N,0,ZP,XP,AREA)
      RETURN
      END
```

```
FUNCTION XELIPS(Z1,Z2,X,Z)
XELIPS=X*SQRT(1.0-((Z-Z1)/(Z2-Z1))**2)
XELIPS=ABS(XELIPS)
RETURN
END
```

```
FUNCTION XELIPP(Z1,Z2,X1,X,Z)
XELIPP=-X1/(Z-Z1)*((Z2-Z1)/X)**2
RETURN
END
```

```
SUBROUTINE CONICS(ZX,ZXP,ICX,ICZ,ISX,ISZ,IB,Z,X)
DIMENSION ZX(1),ZXP(1)
SIG = SIGN(1.,ZX(ICZ)-ZX(ISZ))
H = SIG*(1.414213562*ZX(IB)-ZX(ISX))-ZX(ISZ)
ZZ = Z-ZX(ISZ)
Z1 = ZX(ICZ)-ZX(ISZ)
X1 = ZX(ICX)-ZX(ISX)
CK = (H-Z1)**2/(4.*H*X1*Z1*(H-SIG*X1-Z1))
A = 2.*X1*Z1*ZZ*CK+Z1-ZZ
X = ZX(ISX)+(A-SIG*SQRT((2.*A-Z1+ZZ)*(Z1-ZZ)))/(2.*Z1*Z1*CK)
RETURN
END
```

```
      SUBROUTINE SIGMA2(N,IND,X,Y,SUM)
      DIMENSION X(1),Y(1),SUM(1)
      IS = 1
      SUM(1) = 0.
      DO 40 I=2,N
      ISP = IS+IND
      AREA = (Y(I)+Y(I-1))*(X(I)-X(I-1))/2.0
      SUM(ISP) = SUM(IS)+AREA
   40 IS = ISP
      RETURN
      END
```

```
C****VOLUME CALCULATION PROCEDURE (PHASE II CONTRACT)
      COMMON/ALLQ/C(8,16),NCQ,A(100,3,4)
      COMMON /ONE/X1
      DIMENSION ZX(25),P(25),Q(25),R(25),S(25),T(25),SG(25),ZXP(25)
      DIMENSION NO(14),AAA(500),YYY(500)
      DATA NO/3,20,2,8,3,40,40,3,8,25,5,25,3,3/
      IPRINT=0
      NY=1
      DELTAY=0.
      REWIND 3
      REWIND 2
      READ(3,500) NCQ
      DO 5 J=1,NCQ
      READ(3,201) IC,(C(I,J),I=1,2)
      READ(3,201) IC,(C(I,J),I=3,8)
    5 CONTINUE
  500 FORMAT(14I5)
      YMAXL=-100.
      YMAX=-100.
   10 WRITE(6,600)
  600 FORMAT(' INPUT Y1,Y2,NY,BTMBLN')
      READ(5,*) Y1,Y2,NY,SSS
      XNY=NY-1
      DELTAY=(Y2-Y1)/XNY
      DO 50 LL=1,NY
      XLL=LL-1
      Y=Y1+DELTAY*XLL
      DO 2 I=1,50
      IF((Y.LE.YMAX).AND.(Y.GE.YMAXL)) GO TO 3
      IF(Y.LT.YMAXL) REWIND 2
      YMAXL=YMAX
      READ(2,200) YMAX,N,MOD,RATIO
      IF(MOD.NE.0) MOOD=MOD
      DO 2 K=1,N
      READ(2,201) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
    2 CONTINUE
C**** MOOD=1     FUSELAGE ONLY
C**** MOOD=2     FUSELAGE WITH CANOPY
C**** MOOD=3     FUSELAGE WITH WING
C**** MOOD=4     FUSELAGE WITH WING & CANOPY
    3 IF(MOOD.EQ.1) M=7
      IF(MOOD.EQ.2) M=13
      IF(MOOD.GT.2) M=21
```

```
200 FORMAT(F10.5,2I5,F10.5)
201 FORMAT(I2,5E15.8,F2.0)
    DO 4 I=1,M
    IF((MOOD.EQ.3).AND.(I.GT.7).AND.(I.LT.14)) GO TO 4
    IF((RATIO.GT.0.).AND.((I.EQ.19).OR.(I.EQ.20))) GO TO 4
    CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
  4 CONTINUE
    IF(MOOD.LT.3)GO TO 6
    IF(RATIO.GT.0.) CALL XSUBG(ZX,ZXP,RATIO)
    CALL CONPIC(C,NCQ,Y,ZX(22),ZXP(22),INQ)
    ZX(23)=ZX(15)-SSS
    ZXP(23)=ZXP(15)
  6 CONTINUE
    IF(MOOD.GT.2) M=23
    CALL AREAC(Y,ZX,ZXP,MOOD,NO,AAA(LL))
    YYY(LL)=Y
 50 CONTINUE
    CALL SIGMA2(NY,0,YYY,AAA,AREAT)
    WRITE(6,601) AREAT
601 FORMAT(' VOLUME=',G15.7)
502 FORMAT(F20.0)
700 FORMAT(8G15.7)
701 FORMAT('Y,MOOD/ZX.....',2G15.7)
    GO TO 10
    END
```

```
      SUBROUTINE CONICF(P,Q,R,S,T,SG,X,Y,YP)
      ROOT=SG*SQRT((R*X+S)*X+T)
      Y=P*X+Q+ROOT
      IF((SG.NE.0.).AND.(ROOT.NE.0.)) GO TO 10
      YP=P
      GO TO 20
   10 YP=P+(R*X+S/2.0)/ROOT
   20 RETURN
      END
```

```
      FUNCTION XELIPS(Z1,Z2,X,Z)
      C=1.0-((Z-Z1)/(Z2-Z1))**2
      IF(C.LT.0.) C=0.
      XELIPS=X*SQRT(C)
      XELIPS=ABS(XELIPS)
      RETURN
      END
```

```
FUNCTION XELIPP(Z1,Z2,X1,X,Z)
XELIPP=-X1/(Z-Z1)*((Z2-Z1)/X)**2
RETURN
END
```

```
FUNCTION PELIPS(X1,Z1,X2,Z2,Z2P,X)                    00000890
S=SIGN(1.0,Z2-Z1)                                     00000900
C=X1-X2                                               00000910
DZ=Z2-Z1                                              00000920
A=C*(C+DZ/Z2P)/(2.*C+DZ/Z2P)                          00000930
B2=DZ*DZ-(A-C)*DZ*Z2P                                 00000940
X0=X1-A                                               00000950
D = (X-X0)/A                                          00000960
ROOT=B2*(1.0-((X-X0)/A)**2)                           00000965
IF(ROOT.LT.0.)ROOT=0.                                 00000966
PELIPS=S*SQRT(ROOT)+Z1                                00000970
RETURN                                                00000980
END                                                   00000990
```

```
      SUBROUTINE AREAC(Y,ZX,ZXP,MOOD,NO,AA)
      COMMON/MESH/K1,K2,K3,K4,K5,K6,K7,K8,AL(4)
      COMMON/OK/X0U,Z0U,X0L,Z0L,R0U,R0L,SU,SL
      COMMON /ONE/X1
      DIMENSION XP(200),ZP(200),NO(1),ZX(1),ZXP(1)
      IF(MOOD.LT.3) GO TO 10
      CALL PCRCL(ZX,ZXP,-2.,Y,Z,14,16,19,1,4,2,5,22,D,D,D,D,
     & X2U,Z2U,X0U,Z0U)
      X1U=X1
      CALL PCRCL(ZX,ZXP,-2.,Y,Z,15,17,20,3,6,7,5,23,D,D,D,D,
     & X2L,Z2L,X0L,Z0L)
      X1L=X1
      R0U=SQRT((Z2U-Z0U)**2+(X2U-X0U)**2)
      R0L=SQRT((Z2L-Z0L)**2+(X2L-X0L)**2)
      SU=0
      SL=0
      IF(ZX(5).NE.ZX(20)) SU=(ZX(16)-ZX(14))/(ZX(19)-ZX(5))
      IF(ZX(19).NE.ZX(5)) SL=(ZX(17)-ZX(15))/(ZX(20)-ZX(5))
   10 L=0
      IF(ZX(6).EQ.0.) GO TO 20
      CALL STEP(L,0.,ZX(6),NO,1,1,XP,ZP,ZX)
   20 IF(MOOD.GT.2) Z2=AMIN1(ZX(7),ZX(23))
      IF(MOOD.LT.3) Z2=ZX(7)
      III=0
      IF(ZX(6).EQ.0.) III=1
      CALL STEP(L,ZX(3),Z2,NO,2,III,ZP,XP,ZX)
      IF(MOOD.GT.2) GO TO 30
      IF(ZX(2).LE.ZX(7)) GO TO 70
      CALL STEP(L,ZX(7),ZX(2),NO,3,0,ZP,XP,ZX)
      GO TO 70
   30 IF(ZX(23).LE.ZX(7)) GO TO 35
      CALL STEP(L,ZX(7),ZX(23),NO,3,0,ZP,XP,ZX)
   35 CALL STEP(L,X1L,X2L,NO,4,0,XP,ZP,ZX)
      IF(ZX(20).LE.X2L) GO TO 40
      CALL STEP(L,X2L,ZX(20),NO,5,0,XP,ZP,ZX)
   40 CALL STEP(L,ZX(20),ZX(21),NO,6,0,XP,ZP,ZX)
      CALL STEP(L,ZX(21),ZX(19),NO,7,0,XP,ZP,ZX)
      IF(ZX(19).LE.X2U) GO TO 50
      CALL STEP(L,ZX(19),X2U,NO,8,0,XP,ZP,ZX)
   50 CALL STEP(L,X2U,X1U,NO,9,0,XP,ZP,ZX)
      IF(ZX(22).GE.ZX(2)) GO TO 60
      CALL STEP(L,ZX(22),ZX(2),NO,3,0,ZP,XP,ZX)
   60 IF(MOOD.GT.2) Z1=AMAX1(ZX(2),ZX(22))
```

```
   70 IF(MOOD.LT.3) Z1=ZX(2)
      IF((MOOD.EQ.2).OR.(MOOD.EQ.4)) Z2=ZX(9)
      IF((MOOD.EQ.1).OR.(MOOD.EQ.3)) Z2=ZX(1)
      CALL STEP(L,Z1,Z2,NO,10,0,ZP,XP,ZX)
      IF((ZX(4).EQ.0.).OR.((MOOD.EQ.2).OR.(MOOD.EQ.4))) GO TO 80
      CALL STEP(L,ZX(4),0.,NO,14,0,XP,ZP,ZX)
      GO TO 90
   80 IF((MOOD.EQ.1).OR.(MOOD.EQ.3)) GO TO 90
      CALL STEP(L,ZX(9),ZX(8),NO,11,0,ZP,XP,ZX)
      CALL STEP(L,ZX(8),ZX(10),NO,12,0,ZP,XP,ZX)
      IF(ZX(13).EQ.0) GO TO 90
      CALL STEP(L,ZX(13),0.,NO,13,0,XP,ZP,ZX)
   90 IF(MOOD.GT.2) ZM=ZX(18)
      IF(MOOD.LT.3)ZM=ZX(2)
      XMAX=XP(1)
      IC=1
      DO 100 I=2,L
      IF(XMAX.GT.XP(I)) GO TO 100
      XMAX=XP(I)
      IC=I
  100 CONTINUE
      DO 120 I=1,IC
  120 ZP(I)=ZM-ZP(I)
      DO 130 I=IC,L
  130 ZP(I)=ZP(I)-ZM
      ZP(IC)=0.
      CALL SIGMA2(IC,0,XP,ZP,A1)
      N2=L-IC+1
      CALL SIGMA2(N2,0,XP(IC),ZP(IC),A2)
      AA=A1-A2
      RETURN
      END
```

```
      SUBROUTINE STEP(L,X1,X2,NO,IN,IO,XP,YP,ZX)
      COMMON /OK/XOU,ZOU,XOL,ZOL,ROU,ROL,SU,SL
      COMMON /ONE/X11
      DIMENSION XP(1),YP(1),ZX(1),NO(1)
      N=NO(IN)+IO
      XN=N-IO
      DX=(X2-X1)/XN
      DO 100 I=1,N
      L=L+1
      XI=I-IO
      XP(L)=X1+DX*XI
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14),IN
    1 YP(L)=ZX(3)
      GO TO 100
    2 YP(L)=XELIPS(ZX(7),ZX(3),ZX(5)-ZX(6),XP(L))+ZX(6)
      GO TO 100
    3 YP(L)=ZX(5)
      GO TO 100
    4 ROOT=ROL**2-(XP(L)-XOL)**2
      IF(ROOT.LE.0.) ROOT=0.
      YP(L)=ZOL+SQRT(ROOT)
      GO TO 100
    5 YP(L)=SL*(XP(L)-ZX(5))+ZX(15)
      GO TO 100
    6 YP(L)=PELIPS(ZX(21),ZX(18),ZX(20),ZX(17),SL,XP(L))
      GO TO 100
    7 YP(L)=PELIPS(ZX(21),ZX(18),ZX(19),ZX(16),SU,XP(L))
      GO TO 100
    8 YP(L)=SU*(XP(L)-ZX(19))+ZX(16)
      GO TO 100
    9 ROOT=ROU**2-(XP(L)-XOU)**2
      IF(ROOT.LE.0.) ROOT=0.
      YP(L)=ZOU-SQRT(ROOT)
      GO TO 100
   10 YP(L)=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),XP(L))+ZX(4)
      GO TO 100
   11 QX1=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),ZX(8))+ZX(13)
      QX2=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),ZX(9))+ZX(4)
      Z1P=XELIPP(ZX(11),ZX(10),QX1,ZX(12)-ZX(13),ZX(8))
      Z2P=XELIPP(ZX(2),ZX(1),QX2,ZX(5)-ZX(4),ZX(9))
      DZ1=XP(L)-ZX(8)
      DZ21=ZX(9)-ZX(8)
      A=3.*DZ21-(Z2P+2.*Z1P)*(QX2-QX1)
```

```
      B=-2.*DZ21+(Z2P+Z1P)*(QX2-QX1)
      IF((Z1P.LT.-4.).OR.(Z2P.LT.-4.).OR.(Z1P.GE.0.)) GO TO 22
      GO TO 18
   22 TX=1
      IF(XP(L).EQ.ZX(8)) YP(L)=QX1
      IF(XP(L).EQ.ZX(9)) YP(L)=QX2
      IF((XP(L).EQ.ZX(8)).OR.(XP(L).EQ.ZX(9))) GO TO 100
      XI=QX1
      DO 19 KKK=1,2
      DO 17 J=1,6
      DXR=(XI-QX1)/(QX2-QX1)
      ZI=ZX(8)+Z1P*(XI-QX1)+A*DXR**2+B*DXR**3
      ZIP=Z1P+(2.0*A*DXR+3.0*B*DXR**2)/(QX2-QX1)
      IF(ZIP.NE.0.) GO TO 16
      YP(L)=XI+(QX2-QX1)/100.
      GO TO 17
   16 YP(L)=XI-(ZI-XP(L))/ZIP
      IF(ABS(YP(L)-XI).LE.1.E-5) GO TO 100
   17 XI=YP(L)
      XI=QX2
   19 CONTINUE
      GO TO 100
   18 DQX21=QX2-QX1
      QX1P=1.0/Z1P
      QX2P=1.0/Z2P
      A=3.0*DQX21-(QX2P+2.0*QX1P)*DZ21
      B=-2.0*DQX21+(QX2P+QX1P)*DZ21
      DZR=(XP(L)-ZX(8))/DZ21
      YP(L)=QX1+QX1P*DZ1+A*DZR**2+B*DZR**3
      GO TO 100
   12 YP(L)=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),XP(L))+ZX(13)
      GO TO 100
   13 YP(L)=ZX(10)
      GO TO 100
   14 YP(L)=ZX(1)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PCRCL(ZX,ZXP,X,YI,Z,IZE,IZG,IXG,IZU,IXU,IZM,IXM,
     & IZ1,FX,FY,FZ,F,X2,Z2,X0,Z0)
      COMMON /ONE/X1
      DIMENSION ZX(1),ZXP(1)
      IT=0
      IF((ZX(IZ1).LT.ZX(2)).AND.(ZX(IZ1).GT.ZX(7))) IT=1
      IF((Y.EQ.YI).AND.(ITT.EQ.IT).AND.(IZEP.EQ.IZE)) GO TO 1000
      Y=YI
      ITT=IT
      IZEP=IZE
      XM4=(ZX(IZG)-ZX(IZE))/(ZX(IXG)-ZX(IXM))
      IF(IZG.EQ.16) SS=-SIGN(1.0,XM4)
      IF(IZG.EQ.17) SS=SIGN(1.0,XM4)
      ZUM=ZX(IZU)-ZX(IZM)
      XMU=ZX(IXM)-ZX(IXU)
      IF(IT.NE.1) GO TO 10
      X1=ZX(IXM)
      X1Y=ZXP(IXM)
      XM1=0.
      XM1Y=0.
      XM3P=0.
      XM3PY=0.
      GO TO 20
   10 Z1M=ZX(IZ1)-ZX(IZM)
      Z1MZUM=Z1M/ZUM
      T1=SQRT(1.0-Z1MZUM**2)
      X1=ZX(IXU)+XMU*T1
      X1U=X1-ZX(IXU)
      T2=2.0*Z1MZUM*((ZXP(IZ1)-ZXP(IZM))/ZUM
     & -Z1M*(ZXP(IZU)-ZXP(IZM))/ZUM**2)
      X1Y=-XMU**2/X1U*T2/2.0+T1*(ZXP(IXM)-ZXP(IXU))
     & +ZXP(IXU)
      XMUZUM=XMU/ZUM
      T1=Z1M/X1U
      XM1=XMUZUM**2*T1
      T2=((ZXP(IXM)-ZXP(IXU))/ZUM-XMU*(ZXP(IZU)-ZXP(IZM))
     & /ZUM**2)*T1
      T3=(ZXP(IZ1)-ZXP(IZM))/X1U-Z1M*(X1Y-ZXP(IXU))/X1U**2
      XM1Y=(2.0*T2+XMUZUM*T3)*XMUZUM
      XM3=-1.0/XM1
      XM3P=-1.0/XM3
      XM3Y=XM1Y/XM1**2
      XM3PY=XM3Y/XM3**2
```

```
   20 XM4P=-1.0/XM4
      XGE=ZX(IXG)-ZX(IXM)
      ZGE=ZX(IZG)-ZX(IZE)
      XM4Y=(ZXP(IZG)-ZXP(IZE))/XGE-(ZXP(IXG)-ZXP(IXM))*ZGE/XGE**2
      XM4PY=XM4Y/XM4**2
      DTH=(ATAN(XM3P)+SS*ATAN(XM4P))/2.0
      XM5=TAN(DTH)
      T1=(1.0/COS(DTH))**2
      T2=XM3PY/(1.0+XM3P**2)+XM4PY/(1.0+XM4P**2)
      XM5Y=T1*T2/2.0
      IF(IT.NE.1) GO TO 30
      XP=ZX(IXM)
      ZP=ZX(IZE)
      XPY=ZXP(IXM)
      ZPY=ZXP(IZE)
      GO TO 40
   30 ZE1=ZX(IZE)-ZX(IZ1)
      XM34=XM3-XM4
      T1=(XM4*ZX(IXM)-XM3*X1)
      XP=(ZE1-T1)/XM34
      T2=ZXP(IZE)-ZXP(IZ1)-XM4*ZXP(IXM)-XM4Y*ZX(IXM)+XM3*X1Y+XM3Y*X1
      XPY=(XM34*T2-(ZE1-T1)*(XM3Y-XM4Y))/XM34**2
      ZP=ZX(IZE)+XM4*(XP-ZX(IXM))
      ZPY=ZXP(IZE)+XM4*(XPY-ZXP(IXM))+(XP-ZX(IXM))*XM4Y
   40 XM15=XM1-XM5
      T1=(ZP-ZX(IZ1))-(XM5*XP-XM1*X1)
      T2=ZPY-ZXP(IZ1)-XM5*XPY-XM5Y*XP+XM1*X1Y+XM1Y*X1
      X0=T1/XM15
      X0Y=(XM15*T2-T1*(XM1Y-XM5Y))/XM15**2
      Z0=ZX(IZ1)+XM1*(X0-X1)
      Z0Y=ZXP(IZ1)+XM1*(X0Y-X1Y)+XM1Y*(X0-X1)
      Z0Z1=Z0-ZX(IZ1)
      X0X1=X0-X1
      R2=Z0Z1**2+X0X1**2
      R=SQRT(R2)
 1000 IF(X.LT.-1.) GO TO 2000
      RY=(Z0Z1*(Z0Y-ZXP(IZ1))+X0X1*(X0Y-X1Y))/R
      XX0=X-X0
      ZZ0=Z-Z0
      F=R2-ZZ0**2-XX0**2
      FX=-2.0*XX0
      FY=2.0*(R*RY+XX0*X0Y+ZZ0*Z0Y)
      FZ=-2.0*ZZ0
```

```
      GO TO 50
 2000 CONTINUE
      D=SQRT((X1-XP)**2+(ZX(IZ1)-ZP)**2)
      ANG=ATAN(XM4)
      X2=XP+D*COS(ANG)
      Z2=ZP+XM4*(X2-XP)
   50 RETURN
      END
```

```
      SUBROUTINE CONPIC(C,NC,XP,YP,DYP,IND)
      DIMENSION C(8,4)
      IND=0
      DO 10 I=1,NC
      IF((XP.GE.C(1,I)).AND.(XP.LE.C(2,I))) GO TO 20
   10 CONTINUE
      IND=1
   20 CALL CONICF(C(3,I),C(4,I),C(5,I),C(6,I),C(7,I),C(8,I),XP,YP,DYP)
      RETURN
      END
```

```
      SUBROUTINE XSUBG(ZX,ZXP,R)
      DIMENSION ZX(1),ZXP(1)
      T1=ZX(21)-ZX(5)
      T2=ZX(16)-ZX(18)
      T3=ZX(14)-ZX(18)
      ZX(19)=ZX(21)-T1*T2/(R*T3)
      ZXP(19)=ZXP(21)-(T3*(T2*(ZXP(21)-ZXP(5))+T1*(ZXP(16)-ZXP(18)
     & ))-T1*T2*(ZXP(14)-ZXP(18)))/(R*T3*T3)
      ZX(20)=ZX(19)
      ZXP(20)=ZXP(19)
      RETURN
      END
```

```
      SUBROUTINE SECOND (X,Y,N,XP,YP,DYP,IT)
      DIMENSION X(1),Y(1)
      M = N-1
      ARROW = XP-X(1)
      NI = N/2+1
      IF ((X(NI)-X(1))*ARROW) 2,2,5
2     IP = 2
      GO TO 20
5     DO 10 I=2,M
      IP = I
      IF((XP-X(I))*ARROW) 20,20,10
10    CONTINUE
20    A = XP-X(IP-1)
      B = XP-X(IP)
      C = XP-X(IP+1)
      D = X(IP-1)-X(IP)
      E = X(IP-1)-X(IP+1)
      G = X(IP)-X(IP+1)
      T1 = Y(IP-1)/(D*E)
      T2 = -Y(IP)/(D*G)
      T3 = Y(IP+1)/(E*G)
      YP = T1*B*C+T2*A*C+T3*A*B
      IF (IT.GT.0) DYP=T1*(B+C)+T2*(A+C)+T3*(A+B)
      RETURN
      END
```

```
      SUBROUTINE SIGMA2(N,IND,X,Y,SUM)
      DIMENSION X(1),Y(1),SUM(1)
      IS = 1
      SUM(1) = 0.
      DO 40 I=2,N
      ISP = IS+IND
      AREA = (Y(I)+Y(I-1))*(X(I)-X(I-1))/2.0
      SUM(ISP) = SUM(IS)+AREA
   40 IS = ISP
      RETURN
      END
```

SECTION V
LOGICAL STRUCTURE

1. INTERDEPENDENCE OF SUBROUTINES

The Calling-Called matrix for the program is shown in Figure 28.

2. LISTINGS

Two complete listings of the Surface Fit and Minimum Search Program are given below, following Figure 28. The first listing is for the computer program developed in Phase I using a 6 x 6 Latin Square. The second listing is for the computer program developed in Phase II using a 4 x 4 Latin Square.

Calling \ Called	XXGEN	CROUT	SEARCH	VALUE	XFORM	SIGMA2	CNSTN	BASEA	VOLMD	AREA	KONIC	KONICA	TABLE	READ	CONICF	LINEAR	XELIPL
LATIN1	●	●	●	●	●		●	●	●	●				●			
SEARCH				●			●										
CNSTN				●		●	●			●			●				
AREA												●					
TABLE												●				●	●
READ															●		

(a) LATIN1 Program

Calling \ Called	XXGEN	CROUT	SEARCH	VALUE
LATIN2	●	●	●	●
SEARCH				●

(b) LATIN2 Program

FIGURE 28. CALLING-CALLED MATRIX FOR SURFACE FIT AND MINIMUM SEARCH PROGRAMS

```
      PROGRAM LATIN1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2,TAPE3)
C=======================================================================
C**LATIN SQUARE MINIMUM VALUE SEARCH PROGRAM (5X5 - 6 VARIABLES) 1975
C=======================================================================
      COMMON /XXX/XX(36,29)
      COMMON/CHECK/FS(20),ZZB(8,20),CV(28),VOLB,KIND(20),NS
      COMMON/CHK1/RATIO(20),ZFS(20,20),XFS(20,20),LUM,NFS(20),ISO,
     & ISN,ISNP
      DIMENSION Y(28),C(28),YF(28),B(28,29),SUM(28,29)
      DIMENSION O(6),E(6),XMIN(6),XMAX(6),NP(6),IY(36)
      DIMENSION ZZ(8),DEV(36),IJK(3,11),V(6),SYM(6),TYPE(4)
      DATA TYPE/4HAREA,8H LOCAL-X,6HRADOME,6HVOLUME/
      DATA FSNOSE,INU,NY/50.,3,32/
      DATA IJK/18,2,2,19,7,7,20,2,5,21,3,7,22,2,3,23,5,7/
      M=6
  412 FORMAT(3F10.0,10I5)
  610 FORMAT(I10,16H POINTS REJECTED/I10,16H POINTS ACCEPTED/
     & T12,8HAVERAGE=,G12.5)
  611 FORMAT(19H KIND,FS,ABASE,AMIN,I5,3G12.5)
      WRITE(6,410)
  410 FORMAT(1H1,T20,*LATIN SQUARE MINIMUM VALUE SEARCH PROGRAM*)
      READ(5,400) SYM
      WRITE(6,411) SYM
  411 FORMAT(* VARIABLES ARE LISTED AS. . .*,6A5)
  400 FORMAT(6(3X,A2))
      READ(5,401)IOR,IPRINT
  401 FORMAT(14I5)
      IF((IOR.LT.17).OR.(IOR.GT.28)) IOR=17
      WRITE(6,402) IOR
  402 FORMAT(* ORDER OF FIT. . . .*,I2)
  403 FORMAT(8G9.2)
      IORS=IOR+1
  999 FORMAT(I4,25F4.0)
      KASE=0
      NS=1
C-----------------------------------------------------------------------
   11 READ(5,401) KIND(NS)
      IF(KIND(NS).EQ.4) KASE=1
      IF((KIND(NS).EQ.0).OR.(KIND(NS).GT.4)) GO TO 12
      IT=KIND(NS)
      WRITE(6,404) KIND(NS),TYPE(IT)
  404 FORMAT(13H CONSTRAINT =,I2,1H-,A8)
      GO TO (201,201,202,203),IT
```

```
201 READ(5,403) FS(NS),RATIO(NS)                                        5X500440
    IF((RATIO(NS).NE.0.).OR.(KIND(NS).EQ.1)) GO TO 21                   5X500450
    READ(5,401) NPFS                                                    5X500460
    READ(5,403)(ZFS(K,NS),XFS(K,NS),K=1,NPFS)                           5X500470
    NFS(NS)=NPFS                                                        5X500480
    WRITE(6,405)FS(NS),(K,ZFS(K,NS),XFS(K,NS),K=1,NPFS)                 5X500490
405 FORMAT(* INPUT CROSS SECTION TABLE,F.S.*,G12.5/                     5X500500
   & T10,1HZ,T22,1HX/(I3,2G12.5))                                       5X500510
 21 CALL READ(INU,FS(NS),ZZB(1,NS))                                     5X500520
    GO TO 35                                                            5X500530
202 READ(5,403) FS(NS),RC,RD                                            5X500540
    CALL READ(INU,FS(NS),ZZB(1,NS))                                     5X500550
    ZZB(7,NS)=RC                                                        5X500560
    ZZB(8,NS)=RD                                                        5X500570
    GO TO 35                                                            5X500580
203 READ(5,412) FS1,FS2,ANY,NFSI,LOVE                                   5X500590
    FS(NS)=FS1                                                          5X500600
    RATIO(NS)=ANY                                                       5X500610
    IF(ANY.GT.2.0) VOLB=ANY                                             5X500620
    IF(FS1.LT.FSNOSE) LUM=1                                             5X500630
    IF(FS1.GE.FSNOSE) LIM=2                                             5X500640
    IF(FS1.LE.FSNOSE) FS(NS)=FSNOSE                                     5X500650
    IF((FS2.LE.FSNOSE).OR.(NFSI.LT.2)) LUM=0                            5X500660
    ISO=NS                                                              5X500670
    IS1=NS+1                                                            5X500680
    ISN=ISO+NFSI                                                        5X500690
    IS2=ISN-1                                                           5X500700
    IF(ISN.NE.ISO) FS(ISN)=FS2                                          5X500710
    ISNP=NFSI+1                                                         5X500720
    IC=NFSI-1                                                           5X500730
    NS=ISN                                                              5X500740
    IF((LUM.EQ.0).OR.(ISO.EQ.ISN)) GO TO 33                             5X500750
    IF(LOVE.EQ.0) GO TO 31                                              5X500760
    READ(5,403)(FS(K),K=IS1,IS2)                                        5X500770
    GO TO 33                                                            5X500780
 31 XI=NFSI                                                             5X500790
    DY=(FS(ISN)-FS(ISO))/XI                                             5X500800
    DO 32 K=IS1,IS2                                                     5X500810
    XI=K-IS1+1                                                          5X500820
 32 FS(K)=FS(ISO)+DY*XI                                                 5X500830
 33 DO 34 K=ISO,ISN                                                     5X500840
    CALL READ(INU,FS(K),ZZB(1,K))                                       5X500850
    KIND(K)=4                                                           5X500860
```

```
   34 CONTINUE                                                          5X500870
   35 NS=NS+1                                                           5X500880
      GO TO 11                                                          5X500890
   12 NS=NS-1                                                           5X500900
C-----------------------------------------------------------------------5X500910
    6 CALL XXGEN(IJK,M)                                                 5X500920
      IF(IPRINT.EQ.0) GO TO 8                                           5X500930
      WRITE(6,886)                                                      5X500940
  886 FORMAT(/* ADDITIONAL TERMS ARE. . . .*)                           5X500950
      DO 7 I=1,M                                                        5X500960
      I1=IJK(2,I)-1                                                     5X500970
      I2=IJK(3,I)-1                                                     5X500980
    7 WRITE(6,887) IJK(1,I),SYM(I1),SYM(I2)                             5X500990
  887 FORMAT(2H (,I2,3H)   ,2A5)                                        5X501000
    8 KU=KASE+1                                                         5X501010
      DO 1000 KKK=1,KU                                                  5X501020
      DO 10 I=1,28                                                      5X501030
      DO 10 J=1,29                                                      5X501040
   10 SUM(I,J)=0.0                                                      5X501050
      N=0                                                               5X501060
      DO 20 JY=1,36                                                     5X501070
      READ(5,500) J,ANY,V                                               5X501080
  500 FORMAT(I2,7F10.0)                                                 5X501090
      IF((J.LE.25).OR.(KKK.EQ.2)) GO TO 19                              5X501100
      DO 15 I=2,7                                                       5X501110
   15 XX(J,I)=V(I-1)                                                    5X501120
      DO 16 I=8,11                                                      5X501130
   16 XX(J,I)=XX(J,I-5)**2                                              5X501140
      DO 17 I=12,14                                                     5X501150
   17 XX(J,I)=XX(J,3)*XX(J,I-8)                                         5X501160
      DO 18 I=15,16                                                     5X501170
   18 XX(J,I)=XX(J,4)*XX(J,I-10)                                        5X501180
      XX(J,17)=XX(J,5)*XX(J,6)                                          5X501190
      DO 1 I=1,M                                                        5X501200
      I0=IJK(1,I)                                                       5X501210
      I1=IJK(2,I)                                                       5X501220
      I2=IJK(3,I)                                                       5X501230
    1 XX(J,I0)=XX(J,I1)*XX(J,I2)                                        5X501240
   19 IF(J.EQ.0) GO TO 25                                               5X501250
      N=N+1                                                             5X501260
      XX(J,IORS)=ANY                                                    5X501270
      IY(N)=J                                                           5X501280
      DO 20 K=1,IOR                                                     5X501290
```

```
      DO 20 L=1,IORS                                                    5X501300
   20 SUM(K,L)=SUM(K,L)+XX(J,K)*XX(J,L)                                 5X501310
   25 IF(N.LT.IOR) GO TO 60                                             5X501320
      WRITE(6,607)                                                      5X501330
      IF(KKK.EQ.1) WRITE(6,801)                                         5X501340
      IF(KKK.EQ.2) WRITE(6,802)                                         5X501350
  801 FORMAT(1H+,T17,11H*WAVE DRAG*)                                    5X501360
  802 FORMAT(1H+,T17,8H*VOLUME*)                                        5X501370
  607 FORMAT(15H0INPUT DATA....)                                        5X501380
      DO 55 I=1,N                                                       5X501390
      NN=IY(I)                                                          5X501400
      IF(IPRINT.GT.0) WRITE(6,609) NN,XX(NN,IORS)                       5X501410
   55 CONTINUE                                                          5X501420
  609 FORMAT (I10,G15.7)                                                5X501430
  602 FORMAT(10G12.4)                                                   5X501440
      IF(KKK.EQ.1) CALL CROUT(SUM,B,C,IOR)                              5X501450
      IF(KKK.EQ.2) CALL CROUT(SUM,B,CV,IOR)                             5X501460
      IF((IPRINT.GT.0).AND.(KKK.EQ.1))WRITE(6,803)                      5X501470
  803 FORMAT(T20,*COEFFICIENTS OF WAVE DRAG EQUATION*)                  5X501480
      IF((IPRINT.GT.0).AND.(KKK.EQ.1)) WRITE (6,601) (I,C(I),I=1,IOR)   5X501490
      IF((IPRINT.GT.0).AND.(KKK.EQ.2)) WRITE(6,601)(I,CV(I),I=1,IOR)    5X501500
      DO 30 J=1,N                                                       5X501510
   30 YF(J)=0.0                                                         5X501520
      DO 40 I=1,N                                                       5X501530
      DO 40 K=1,IOR                                                     5X501540
      IF(KKK.EQ.1) YF(I)=YF(I)+XX(I,K)*C(K)                             5X501550
      IF(KKK.EQ.2) YF(I)=YF(I)+XX(I,K)*CV(K)                            5X501560
   40 CONTINUE                                                          5X501570
C-----------------------------------------------------------------------15X501580
      SUMS=0.                                                           5X501590
      SUMSUM=0.                                                         5X501600
      DO 45 K=1,N                                                       5X501610
      NN=IY(K)                                                          5X501620
      DEV(NN)=YF(NN)-XX(NN,IORS)                                        5X501630
      SUMSUM=SUMSUM+XX(NN,IORS)                                         5X501640
   45 SUMS=SUMS+(DEV(NN))**2                                            5X501650
      XNY=N                                                             5X501660
      STD=SQRT(SUMS/XNY)                                                5X501670
      AVG=SUMSUM/XNY                                                    5X501680
      IF(IPRINT.EQ.0) GO TO 1000                                        5X501690
      WRITE(6,699) STD                                                  5X501700
  699 FORMAT(9H STD DEV=,G12.5)                                         5X501710
      IF((IPRINT.GT.0).AND.(KKK.EQ.1)) WRITE(6,804)                     5X501720
```

```
  804 FORMAT(T20,*FITTED WAVE DRAG*)                                        5X501730
      WRITE(6,601)  (I,YF(I),I=1,N)                                         5X501740
      WRITE(6,710)                                                          5X501750
      WRITE(6,805)                                                          5X501760
  805 FORMAT(T20,*ERROR OF FIT*)                                            5X501770
      WRITE(6,601)(I,DEV(I),I=1,N)                                          5X501780
 1000 CONTINUE                                                              5X501790
C-----------------------------------------------------------------------L 5X501800
      IF((KASE.EQ.1).AND.(RATIO(ISO).LE.2.0))  VOLB=XX(26,IORS)             5X501810
  601 FORMAT(5(1X,1H(,I2,1H),G15.7,1X))                                     5X501820
   50 WRITE(6,603)                                                          5X501830
  603 FORMAT(* INPUT NO. OF PTS.*)                                          5X501840
  604 FORMAT(* INPUT LOWER LIMITS*)                                         5X501850
      READ(5,401) NP                                                        5X501860
      IF(EOF(5)) 1122,59                                                    5X501870
   59 WRITE(6,401) NP                                                       5X501880
      WRITE(6,604)                                                          5X501890
      READ(5,403) O                                                         5X501900
      WRITE(6,403) O                                                        5X501910
      WRITE(6,605)                                                          5X501920
  605 FORMAT(* INPUT UPPER LIMITS*)                                         5X501930
      READ(5,403) E                                                         5X501940
      WRITE(6,403) E                                                        5X501950
C-----------------------------------------------------------------------)5X501960
      CALL SEARCH(O,E,NP,C,IC,NO,YSUM,XMIN,XMAX,YMIN,YMAX,YAVG,IOR,IJK,M5X501970
     & )                                                                    5X501980
C-----------------------------------------------------------------------)5X501990
  711 FORMAT(/T7,2HCD,T22,6(A2,13X))                                        5X502000
      WRITE(6,610) NO,IC,YAVG                                               5X502010
      WRITE(6,806)                                                          5X502020
  806  FORMAT(1H0,T10,*MINIMUM DRAG CONFIGURATION*)                         5X502030
      WRITE(6,711) SYM                                                      5X502040
      WRITE(6,606) YMIN,XMIN                                                5X502050
      IF(NS.EQ.0) GO TO 50                                                  5X502060
      DO 76 K=1,NS                                                          5X502070
      CALL XFORM(XMIN,ZZB(1,K),ZZ)                                          5X502080
      CALL BASEA(XX1,IX2,IX3,IX4,IX5,ABASE,K,VMIN)                          5X502090
      CALL AREA(ZZ,AMIND,XMIN)                                              5X502100
      WRITE(6,710)                                                          5X502110
  710 FORMAT(1H )                                                           5X502120
      WRITE(6,611) KIND(K),FS(K),ABASE,AMIND                                5X502130
   76 CONTINUE                                                              5X502150
   77 IF(KASE.NE.1) GO TO 50                                                5X502160
```

```
      CALL CNSTN(XMIN,INDEX,IOR,IJK,M,XX1,IX2,XX3)                  5X502170
      CALL VOLMD(XX1,IX2,IX3,IX4,IX5,XX6,IX7,VMIN)                  5X502180
      VNOSE=VALUE(XMIN,CV,IOR,IJK,M)                                5X502190
      WRITE(6,710)                                                  5X502200
      WRITE(6,620) VOLB,VMIN,VNOSE                                  5X502210
  620 FORMAT(*0VOLUME - BASELINE,MIN. CONF.,NOSE...*,3G15.7)        5X502215
      GO TO 50                                                      5X502220
   60 WRITE(6,608) N                                                5X502230
  608 FORMAT(* INSUFFICIENT NO. OF INPUT DATA*/* N=*,I2)            5X502240
  606 FORMAT(8G15.7)                                                5X502250
 1122 STOP                                                          5X502260
      END                                                           5X502270
```

```
C******************************************************************XXGEN5X502280
      SUBROUTINE XXGEN(IJK,M)                                           5X502290
      COMMON /XXX/XX(36,29)                                             5X502300
      DIMENSION JS(5),XB(5,5),XL(28),IJK(3,11)                          5X502310
      EQUIVALENCE (XB(1,1),XX(1,2))                                     5X502320
      EQUIVALENCE (XB(1,1),XL(1))                                       5X502330
      DATA JS/3,4,0,1,2/                                                5X502340
      DO 5 I=1,36                                                       5X502350
    5 XX(I,1)=1.0                                                       5X502360
      DO 10 J=1,5                                                       5X502370
      DO 10 I=1,5                                                       5X502380
      XI=I-1                                                            5X502390
      XB(I,J)=-2+XI                                                     5X502400
   10 CONTINUE                                                          5X502410
      DO 20 J=1,5                                                       5X502420
      I1=1+(J-1)*5                                                      5X502430
      I2=I1+4                                                           5X502440
      DO 20 I=I1,I2                                                     5X502450
      XX(I,7)=XB(J,1)                                                   5X502460
   20 CONTINUE                                                          5X502470
      DO 30 J=1,5                                                       5X502480
      I1=1+(J-1)*5                                                      5X502490
      I2=I1+4                                                           5X502500
      JUMP=JS(J)                                                        5X502510
      DO 30 K=3,6                                                       5X502520
      XK=K-2                                                            5X502530
      L1=JUMP*XK                                                        5X502540
      IC=0                                                              5X502550
      DO 30 I=I1,I2                                                     5X502560
      IC=IC+1                                                           5X502570
      XX(I,K)=XL(L1+IC)                                                 5X502580
   30 CONTINUE                                                          5X502590
      DO 50 K=8,11                                                      5X502600
      DO 50 I=1,25                                                      5X502610
      XX(I,K)=XX(I,K-5)**2                                              5X502620
   50 CONTINUE                                                          5X502630
      DO 60 K=12,14                                                     5X502640
      DO 60 I=1,25                                                      5X502650
   60 XX(I,K)=XX(I,3)*XX(I,K-8)                                         5X502660
      DO 70 K=15,16                                                     5X502670
      DO 70 I=1,25                                                      5X502680
   70 XX(I,K)=XX(I,4)*XX(I,K-10)                                        5X502690
      DO 80 I=1,25                                                      5X502700
```

```
80 XX(I,17)=XX(I,5)*XX(I,6)
   IF(M.EQ.0) RETURN
   DO 40 I=1,25
   DO 40 J=1,M
   I0=IJK(1,J)
   I1=IJK(2,J)
   I2=IJK(3,J)
   XX(I,I0)=XX(I,I1)*XX(I,I2)
40 CONTINUE
   RETURN
   END
```

```
C*******************************************************************CROUT5X502820
      SUBROUTINE CROUT(A,AUX,B,N)                                       5X502830
      DIMENSION A(28,29),AUX(28,29),B(1)                                5X502840
C     A IS GIVEN MATRIX, AUX IS AUXILARY MATRIX, B IS SOLUTION VECTOR   5X502850
      NN = N+1                                                          5X502860
      DO 1 J=1,N                                                        5X502870
      DO 1 K=1,NN                                                       5X502880
 1    AUX(J,K) = A(J,K)                                                 5X502890
      DO 2 J=2,NN                                                       5X502900
 2    AUX(1,J) = A(1,J)/AUX(1,1)                                        5X502910
      DO 5 J=2,N                                                        5X502920
      K = J-1                                                           5X502930
      DO 5 JJ=J,N                                                       5X502940
      JJP1 = JJ+1                                                       5X502950
      DO 4 L=1,K                                                        5X502960
      AUX(JJ,J) = AUX(JJ,J) - AUX(L,J)*AUX(JJ,L)                        5X502970
 4    AUX(J,JJP1) = AUX(J,JJP1) - AUX(L,JJP1)*AUX(J,L)                  5X502980
 5    AUX(J,JJP1) = AUX(J,JJP1)/AUX(J,J)                                5X502990
      DO 6 J=1,N                                                        5X503000
 6    B(J) = AUX(J,NN)                                                  5X503010
      DO 7 J=2,N                                                        5X503020
      K = NN-J                                                          5X503030
      M = J-1                                                           5X503040
      DO 7 L=1,M                                                        5X503050
      NNL = NN-L                                                        5X503060
 7    B(K) = B(K) - AUX(K,NNL)*B(NNL)                                   5X503070
      RETURN                                                            5X503080
      END                                                               5X503090
```

```
C****************************************************************SEARCH5X503100
      SUBROUTINE SEARCH(O,E,NP,C,IC,NO,SUM,XMIN,XMAX,YMIN,YMAX,YAVG,IOR,5X503110
     & IJK,M)                                                           5X503120
      DIMENSION O(6),E(6),NP(6),D(6),X(6),C(1),XMAX(6),XMIN(6),IJK(3,11)5X503130
      DO 10 I=1,6                                                       5X503140
      IF((NP(I).LE.0))NP(I)=1                                           5X503150
      XN=NP(I)-1                                                        5X503160
      D(I)=0.                                                           5X503170
      IF(XN.NE.0.) D(I)=(E(I)-O(I))/XN                                  5X503180
   10 CONTINUE                                                          5X503190
      SUM=0.                                                            5X503200
      NO=0                                                              5X503210
      IC=0                                                              5X503220
      YMAX=-1.0E50                                                      5X503230
      YMIN=1.0E50                                                       5X503240
      N1=NP(1)                                                          5X503250
      DO 1000 I1=1,N1                                                   5X503260
      XI1=I1-1                                                          5X503270
      X(1)=O(1)+XI1*D(1)                                                5X503280
      N2=NP(2)                                                          5X503290
      DO 1000 I2=1,N2                                                   5X503300
      XI2=I2-1                                                          5X503310
      X(2)=O(2)+XI2*D(2)                                                5X503320
      N3=NP(3)                                                          5X503330
      DO 1000 I3=1,N3                                                   5X503340
      XI3=I3-1                                                          5X503350
      X(3)=O(3)+XI3*D(3)                                                5X503360
      N4=NP(4)                                                          5X503370
      DO 1000 I4=1,N4                                                   5X503380
      XI4=I4-1                                                          5X503390
      X(4)=O(4)+XI4*D(4)                                                5X503400
      N5=NP(5)                                                          5X503410
      DO 1000 I5=1,N5                                                   5X503420
      XI5=I5-1                                                          5X503430
      X(5)=O(5)+XI5*D(5)                                                5X503440
      N6=NP(6)                                                          5X503450
      DO 1000 I6=1,N6                                                   5X503460
      XI6=I6-1                                                          5X503470
      X(6)=O(6)+XI6*D(6)                                                5X503480
      CALL CNSTN(X,INDEX,IOR,IJK,M,XX1,IX2,XX3)                         5X503490
      IF(INDEX.EQ.0) GO TO 50                                           5X503500
      NO=NO+1                                                           5X503510
      GO TO 1000                                                        5X503520
```

```
   50 Y=VALUE(X,C,IOR,IJK,M)                                   5X503530
      IC=IC+1                                                  5X503540
      SUM=SUM+Y                                                5X503550
      IF (Y.LT.YMAX) GO TO 200                                 5X503560
      DO 100 I=1,6                                             5X503570
  100 XMAX(I)=X(I)                                             5X503580
      YMAX=Y                                                   5X503590
  200 IF(Y.GT.YMIN) GO TO 1000                                 5X503600
      DO 300 I=1,6                                             5X503610
  300 XMIN(I)=X(I)                                             5X503620
      YMIN=Y                                                   5X503630
 1000 CONTINUE                                                 5X503640
      XIC=IC                                                   5X503650
      YAVG=SUM/XIC                                             5X503660
      RETURN                                                   5X503670
      END                                                      5X503680
```

```
C******************************************************************VALUE5X503690
      FUNCTION VALUE(X,C,IOR,IJK,M)                                     5X503700
      DIMENSION X(1),C(1),Y(36),IJK(3,11)                               5X503710
      VALUE=C(1)                                                        5X503720
      DO 10 I=2,7                                                       5X503730
   10 Y(I)=X(I-1)                                                       5X503740
      DO 20 I=8,11                                                      5X503750
   20 Y(I)=X(I-6)**2                                                    5X503760
      DO 30 I=12,14                                                     5X503770
   30 Y(I)=X(2)*X(I-9)                                                  5X503780
      DO 40 I=15,16                                                     5X503790
   40 Y(I)=X(3)*X(I-11)                                                 5X503800
      Y(17)=X(4)*X(5)                                                   5X503810
      IF(M.EQ.0) GO TO 60                                               5X503820
      DO 50 I=1,M                                                       5X503830
      I0=IJK(1,I)                                                       5X503840
      I1=IJK(2,I)                                                       5X503850
      I2=IJK(3,I)                                                       5X503860
   50 Y(I0)=Y(I1)*Y(I2)                                                 5X503870
   60 DO 70 I=2,IOR                                                     5X503880
   70 VALUE=VALUE+C(I)*Y(I)                                             5X503890
      RETURN                                                            5X503900
      END                                                               5X503910
```

```
C******************************************************************XFORM5X503920
      SUBROUTINE XFORM(X,ZZB,ZZ)                                        5X503930
      DIMENSION X(6),ZZB(8),ZZ(6)                                       5X503940
      DATA COS45/.707106781/                                            5X503950
      ZZ(1)=(.0625*(X(6)+2.)+.9)*ZZB(1)                                 5X503960
      ZZ(2)=(.15*(X(5)+2.)+.6)*ZZB(2)                                   5X503970
      ZZ(3)=.375*(X(3)+2.)*ZZB(3)                                       5X503980
      ZZ(4)=-3.75*(X(4)+2.)+5.0+ZZB(4)                                  5X503990
      C1M=ZZB(2)-ZZB(1)                                                 5X504000
      C1L=ZZB(4)-ZZB(3)                                                 5X504010
      C2B=(ZZB(4)-ZZB(1))*COS45                                         5X504020
      C1B=AMIN1(C1M,C1L)*COS45                                          5X504030
      DC21B=C2B-C1B                                                     5X504040
      XBB=(-ZZB(5)-C1B)/DC21B                                           5X504050
      C1M=ZZ(2)-ZZ(1)                                                   5X504060
      C1L=ZZ(4)-ZZ(3)                                                   5X504070
      C2=(ZZ(4)-ZZ(1))*COS45                                            5X504080
      C1=AMIN1(C1M,C1L)*COS45                                           5X504090
      DC21=C2-C1                                                        5X504100
      REF=C1+DC21*XBB                                                   5X504110
      IF(X(2).EQ.0.) ZZ(5)=REF                                          5X504120
      IF(X(2).LT.0.)ZZ(5)=REF+(REF-C1)/3.*X(2)                          5X504130
      IF(X(2).GT.0.)ZZ(5)=REF+(C2-REF)/3.*X(2)                          5X504140
      ZZ(5)=-ZZ(5)                                                      5X504150
      ZZ(6)=ZZB(6)                                                      5X504160
      ZZ(7)=ZZB(7)                                                      5X504170
      ZZ(8)=ZZB(8)                                                      5X504180
      RETURN                                                            5X504190
      END                                                               5X504200
```

```
C*********************************************************SIGMA25X504210
      SUBROUTINE SIGMA2(N,IND,X,Y,SUM)                           5X504220
      DIMENSION X(1),Y(1),SUM(1)                                 5X504230
      IS = 1                                                     5X504240
      SUM(1) = 0.                                                5X504250
      DO 40 I=2,N                                                5X504260
      IF(I-N) 20,10,10                                           5X504270
   10 X1=X(I-1)                                                  5X504280
      X2=X(I)                                                    5X504290
      GO TO 30                                                   5X504300
   20 X1=X(I-1)                                                  5X504310
      X2=X(I)                                                    5X504320
      X3=X(I+1)                                                  5X504330
      D=X1-X2                                                    5X504340
      E=X1-X3                                                    5X504350
      G=X2-X3                                                    5X504360
      T1=Y(I-1)/(D*E)                                            5X504370
      T2=-Y(I)/(D*G)                                             5X504380
      T3=Y(I+1)/(E*G)                                            5X504390
      A=(T1+T2+T3)/3.0                                           5X504400
      B=-(T1*(X2+X3)+T2*(X1+X3)+T3*(X1+X2))/2.0                  5X504410
      C=T1*X2*X3+T2*X1*X3+T3*X1*X2                               5X504420
   30 ISP=IS+IND                                                 5X504430
      AREA=(((A*X2)+B)*X2+C)*X2-(((A*X1)+B)*X1+C)*X1             5X504440
      SUM(ISP) = SUM(IS)+AREA                                    5X504450
40    IS = ISP                                                   5X504460
      RETURN                                                     5X504470
      END                                                        5X504480
```

```
C*****************************************************************CNSINSX504490
      SUBROUTINE CNSTN(X,INDEX,IOR,IJK,M,ABASE,K,VMIN)                  5X504500
      COMMON/CHECK/FS(20),ZZB(8,20),CV(28),VOLB,KIND(20),NS             5X504510
      COMMON/CHK1/RATIO(20),ZFS(20,20),XFS(20,20),LUM,NFS(20),          5X504520
     & ISO,ISN,ISNP                                                     5X504530
      DIMENSION X(6),ZZ(8),AB(20),AA(20),LIGHT(20),IJK(3,11)            5X504540
      DATA NP,LIGHT/20,20*0/                                            5X504550
      INDEX=0                                                           5X504560
      IF(NS.EQ.0) GO TO 2000                                            5X504570
      DO 100 I=1,NS                                                     5X504580
      CALL XFORM(X,ZZB(1,I),ZZ)                                         5X504590
      IT=KIND(I)                                                        5X504600
      GO TO (10,20,20,30),IT                                            5X504610
C***AREA CONSTRAINT                                                     5X504620
   10 IF(LIGHT(I).EQ.1) GO TO 15                                        5X504630
      CALL AREA(ZZB(1,I),AB(I),X)                                       5X504640
      IF(RATIO(I).LE.2.0) AB(I)=AB(I)*RATIO(I)                          5X504650
      IF(RATIO(I).GT.2.0) AB(I)=RATIO(I)                                5X504660
   15 CALL AREA(ZZ,A,X)                                                 5X504670
      IF(A-AB(I)) 1000,100,100                                          5X504680
C***LOCAL FUSELAGE WIDTH OR CIRCULAR RADOM                              5X504690
   20 CALL TABLE(ZZB(1,I),ZZ,NP,INDEX,X,KIND(I),ZFS(1,I),XFS(1,I),      5X504700
     * NFS(I))                                                          5X504710
      IF(INDEX.EQ.1) GO TO 2000                                         5X504740
C***VOLUME CONSTRAINT                                                   5X504750
   30 IF((I.NE.ISO).OR.(LUM.GT.2)) GO TO 40                             5X504760
      VOLUME=VALUE(X,CV,IOR,IJK,M)                                      5X504770
   40 IF(LUM.LT.1) GO TO 60                                             5X504780
      CALL AREA(ZZ,AA(I),X)                                             5X504790
      IF(LIGHT(I).EQ.0) CALL AREA(ZZB(1,I),AB(I),X)                     5X504800
   50 IF(I.NE.ISN) GO TO 100                                            5X504810
      CALL SIGMA2(ISNP,0,FS(ISO),AA(ISO),VOLADD)                        5X504820
      VOLUME=VOLUME+VOLADD                                              5X504830
      IF(LIGHT(I).EQ.1) GO TO 60                                        5X504840
      LIGHT(I)=1                                                        5X504850
      CALL SIGMA2(ISNP,0,FS(ISO),AB(ISO),VOLADD)                        5X504860
      VOLB=VOLB+VOLADD                                                  5X504870
      IF(RATIO(ISO).LE.2.0) VOLB=VOLB*RATIO(ISO)                        5X504880
   60 IF(VOLUME-VOLB) 1000,100,100                                      5X504890
  100 CONTINUE                                                          5X504900
      GO TO 2000                                                        5X504910
 1000 INDEX=1                                                           5X504920
 2000 RETURN                                                            5X504930
```

```
C-----------------------------------------------------------------------BASEA5X504940
      ENTRY BASEA                                                              5X504950
      ABASE=AB(K)                                                              5X504960
      RETURN                                                                   5X504970
C-----------------------------------------------------------------------VOLMD5X504980
      ENTRY VOLMD                                                              5X504990
      VMIN=VOLUME                                                              5X505000
      RETURN                                                                   5X505010
      END                                                                      5X505020
```

```
C**********************************************************************AREA5X505030
      SUBROUTINE AREA(ZZ,A,X)                                          5X505040
      DIMENSION ZZ(1),X(1)                                             5X505050
      DATA PI/3.14159265358979/                                        5X505060
      X1=ZZ(1)-ZZ(7)                                                   5X505070
      X2=ZZ(6)-ZZ(2)                                                   5X505080
      IF(ZZ(8).NE.0.) GO TO 10                                         5X505090
      AE=(PI*X1/4.0+ZZ(7))*X2                                          5X505100
      GO TO 20                                                         5X505110
   10 T=(ZZ(8)-ZZ(2))/X2                                               5X505120
      ASINT=ASIN(T)                                                    5X505130
      ROOT=SQRT(1.0-T*T)                                               5X505140
      AE=X1*X2/2.0*(T*ROOT+ASINT)+(ZZ(8)-ZZ(2))*ZZ(7)                  5X505150
   20 CALL KONICA(ZZ(3),ZZ(4),ZZ(1),ZZ(2),ZZ(5),XX1,XX2,AK,ZZ,X)       5X505160
      A=AE+AK                                                          5X505170
      RETURN                                                           5X505180
      END                                                              5X505190
```

```
C*******************************************************************KONIC5X505200
      SUBROUTINE KONIC(XL,ZL,XM,ZM,B,Z,X,ARE,ZZX,XX)                   5X505210
      DIMENSION SUM(2),ZZX(8),XX(6)                                    5X505220
      ID=1                                                             5X505230
      GO TO 10                                                         5X505240
      ENTRY KONICA                                                     5X505250
      ID=2                                                             5X505260
   10 SIG=SIGN(1.0,ZL-ZM)                                              5X505270
      H=SIG*(1.414213562*B-XM)-ZM                                      5X505280
      Z1=ZL-ZM                                                         5X505290
      X1=XL-XM                                                         5X505300
      CK=(H-Z1)**2/(4.*H*X1*Z1*(H-SIG*X1-Z1))                          5X505310
      IF(ID.EQ.2) GO TO 20                                             5X505320
      ZZ=Z-ZM                                                          5X505330
      A=2.*X1*Z1*7Z*CK+Z1-ZZ                                           5X505340
      X=XM+(A-SIG*SQRT((2.*A-Z1+ZZ)*(Z1-ZZ)))/(2.*Z1*Z1*CK)            5X505350
      RETURN                                                           5X505360
   20 Z=ZL-ZM                                                          5X505370
      C=4.0*X1*Z1*CK-1.                                                5X505380
      DO 30 I=1,2                                                      5X505390
      Z1PCZ=Z1+C*Z                                                     5X505400
      Z1MZ=Z1-Z                                                        5X505410
      IF(C)24,22,22                                                    5X505420
   22 ARG=(C*Z1MZ)/Z1PCZ                                               5X505430
      IF(ARG.LT.0.) GO TO 35                                           5X505440
      E=2./SQRT(C)*ATAN(SQRT(ARG))                                     5X505450
      GO TO 26                                                         5X505460
   24 ARG1=C*Z1PCZ                                                     5X505470
      ARG2=-Z1MZ                                                       5X505480
      IF((ARG1.LT.0.).OR.(ARG2.LT.0.)) GO TO 35                        5X505490
      E=2./SQRT(-C)*ALOG(SQRT(ARG1)-C*SQRT(ARG2))                      5X505500
   26 ARG3=Z1PCZ*Z1MZ                                                  5X505510
      IF(ARG3.LT.0.) GO TO 35                                          5X505520
      E=(((C+1.)*Z1-2.*C*Z1MZ)*SQRT(ARG3)+                             5X505530
     & (C+1.)**2*Z1**2*E/2.)/(4.*C)                                    5X505540
      SUM(I)=XM*Z+((Z1+(C-1.)*Z/4.)*Z-SIG*E)/(2.*Z1*Z1*CK)             5X505550
   30 Z=0.                                                             5X505560
      ARE=SUM(2)-SUM(1)                                                5X505570
      GO TO 50                                                         5X505580
   35 WRITE(6,600) XX                                                  5X505590
      ARE=0.                                                           5X505600
  600 FORMAT(* NEGATIVE SQRT ENCOUNTERED IN KONIC*/* X=*,6G12.5)       5X505610
   50 CONTINUE                                                         5X505620
```

```
      RETURN                                        5X505630
      END                                           5X505640
```

```
C************************************************************************TABLE5X505650
      SUBROUTINE TABLE(ZZB,ZZ,NP,INDEX,XX,KIND,ZFS,XFS,NFS)             5X505660
      DIMENSION ZZB(6),ZZ(6),ZFS(1),XFS(1)                              5X505670
      XB=0.                                                             5X505680
      IF(KIND.EQ.3) GO TO 20
      Z1=AMAX1(ZFS(1),ZZ(4))                                            5X505700
      Z2=AMIN1(ZFS(NFS),ZZ(6))                                          5X505710
      GO TO 30                                                          5X505720
   20 Z1=AMAX1((ZZB(7)-ZZB(8)),ZZ(4))                                   5X505730
      Z2=AMIN1((ZZB(7)-ZZB(8)),ZZ(6))                                   5X505740
   30 ZNP=NP-1                                                          5X505750
      DZ=(Z2-Z1)/ZNP                                                    5X505760
      DO 100 I=1,NP                                                     5X505770
      XI=I-1                                                            5X505780
      Z=Z1+XI*DZ                                                        5X505790
      IF(KIND.EQ.3) GO TO 32                                            5X505800
      XB=LINEAR(ZFS,XFS,NFS,Z)                                          5X505810
   32 IF(Z.GE.ZZ(2))X=XELIPS(ZZ(2),ZZ(6),ZZ(1),Z)                       5X505820
   34 IF(Z.LT.ZZ(2))CALL KONIC(ZZ(3),ZZ(4),ZZ(1),ZZ(2),ZZ(5),Z,X        5X505860
     * ,XX1,XX2,XX3)                                                    5X505870
      IF(KIND.NE.3) GO TO 50                                            5X505880
   40 ROOT=ZZB(8)**2-(Z-ZZB(7))**2                                      5X505890
      IF(ROOT.LE.0.) GO TO 50                                           5X505900
      XB=SQRT(ROOT)                                                     5X505910
   50 IF(X.LT.XB) GO TO 200                                             5X505920
  100 CONTINUE                                                          5X505930
      GO TO 300                                                         5X505940
  200 INDEX=1                                                           5X505950
  300 RETURN                                                            5X505960
      END                                                               5X506060
```

```
C****************************************************************XELIPS 5X506070
      FUNCTION XELIPS(Z1,Z2,X,Z)                                        5X506080
      XELIPS=X*SQRT(1.-((Z-Z1)/(Z2-Z1))**2)                             5X506090
      XELIPS=ABS(XELIPS)                                                5X506100
      RETURN                                                            5X506110
      END                                                               5X506120
```

```
C**********************************************************************READ 5X506130
      SUBROUTINE READ(INU,Y,ZZ)                                          5X506140
      DIMENSION ZX(13),P(13),Q(13),R(13),S(13),T(13),SG(13),ZZ(8)        5X506150
      DATA YMAXL,YMAX/-100.,-100./                                       5X506160
      DO 2 I=1,50                                                        5X506170
      IF((Y.LE.YMAX).AND.(Y.GE.YMAXL)) GO TO 3                           5X506180
      IF(Y.LT.YMAXL) REWIND INU                                          5X506190
      YMAXL=YMAX                                                         5X506200
      READ(INU,200) YMAX,N                                               5X506210
      M=7                                                                5X506220
      DO 2 K=1,N                                                         5X506230
      READ(INU,201) L,P(L),Q(L),R(L),S(L),T(L),SG(L)                     5X506240
      IF(L.GT.7) M=13                                                    5X506250
    2 CONTINUE                                                           5X506260
  200 FORMAT(F10.5,I5)                                                   5X506270
  201 FORMAT(I2,5E15.8,F2.0)                                             5X506280
    3 DO 4 I=1,7                                                         5X506290
      CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),D)              5X506300
    4 CONTINUE                                                           5X506310
      IF(M.EQ.7) GO TO 6                                                 5X506320
      DO 5 I=8,13                                                        5X506330
    5 CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),D)              5X506340
    6 ZZ(1)=ZX(5)                                                        5X506350
      ZZ(2)=ZX(2)                                                        5X506360
      ZZ(3)=ZX(6)                                                        5X506370
      ZZ(4)=ZX(3)                                                        5X506380
      ZZ(5)=ZX(7)                                                        5X506390
      ZZ(6)=ZX(1)                                                        5X506400
      ZZ(7)=ZX(4)                                                        5X506410
      ZZ(8)=ZX(8)                                                        5X506420
      IF(M.EQ.7) ZZ(8)=0.0                                               5X506430
      RETURN                                                             5X506440
      END                                                                5X506450
```

```
C****************************************************************CONICF 5X506460
      SUBROUTINE CONICF(P,Q,R,S,T,SG,X,Y,YP)                            5X506470
      ROOT=SG*SQRT((R*X+S)*X+T)                                         5X506480
      Y=P*X+Q+ROOT                                                      5X506490
      IF(SG.NE.0.) GO TO 10                                             5X506500
      YP=P                                                              5X506510
      GO TO 20                                                          5X506520
   10 YP=P+(R*X+S/2.0)/ROOT                                             5X506530
   20 RETURN                                                            5X506540
      END                                                               5X506550
```

5X506560

```
      FUNCTION LINEAR(X,Y,N,XP)
      DIMENSION X(100),Y(100)
      DO 30 I=2,N
      IP=I
      IF((XP-X(I)).LE.0.) GO TO 40
   30 CONTINUE
   40 LINEAR=Y(IP-1)+(Y(IP)-Y(IP-1))*(XP-X(IP-1))/(X(IP)-X(IP-1))
      RETURN
      END
```

```
      PROGRAM LATIN2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C=========================================================================
C***LATIN SQUARE MINIMUM VALUE SEARCH PROGRAM (3X3 - 4 VARIABLES)
C=========================================================================
      COMMON /XXX/XX(36,29)
      COMMON/CHECK/FS(20),ZZB(8,25),CV(28),VOLB,KIND(20),NS
      COMMON/CHK1/RATIO(20),ZFS(20,20),XFS(20,20),LUM,NFS(20),ISO,
     & ISN,ISNP
      DIMENSION Y(28),C(28),YF(28),B(28,29),SUM(28,29)
      DIMENSION O(4),E(4),XMIN(4),XMAX(4),NP(4),IY(36)
      DIMENSION ZZ(8),DEV(36),IJK(3,11),V(4),SYM(4)
      DATA FSNOSE,INU,NY/430.,3,32/
      DATA IJK/10,3,3/
      M=1
      READ(5,400) (SYM(I),I=2,4),SYM(1)
  400 FORMAT(4(3X,A2))
      DO 1010 I=1,36
      DO 1010 J=1,29
 1010 XX(I,J)=0.
      WRITE(6,888)
  888 FORMAT(* INPUT ORDER OF FIT - 8 THRU 14*)
      READ(5,401) IOR,IPRINT
      WRITE(6,466) IOR,IPRINT
  466 FORMAT(*  IOR,IPRINT*,2I10)
  401 FORMAT(14I5)
      IF((IOR.LT.8).OR.(IOR.GT.14)) IOR=8
      IORS=IOR+1
  999 FORMAT(I4,25F4.0)
      KASE=0
      NS=1
      ISO =1
C-------------------------------------------------------------------------
   11 READ(5,490) KINDT,VOLBIN
  490 FORMAT(I5,F10.0)
      VOLB=VOLBIN
      IF(KINDT.NE.0) WRITE(6,722) VOLB
  722 FORMAT(*0VOLUME CONSTRAINT, VOLUME*,G16.7,*GREATER THAN BASELINE*
     + )
      IF(KINDT.EQ.0) WRITE(6,721)
  721 FORMAT(*0NO CONSTRAINT*)
 1113 IF(KINDT.EQ.0) KIND(NS)=4
      IF(KINDT.NE.0) KIND(NS)=3
      IF(KIND(NS).EQ.3) KASE=1
```

```
  412 FORMAT(3F10.0,10I5)                                                  3X300440
C-------------------------------------------------------------------------3X300450
    6 CALL XXGEN(IJK,M)                                                    3X300460
      IF(IPRINT.EQ.0) GO TO 8                                              3X300470
      WRITE(6,886)                                                         3X300480
  886 FORMAT(/* ADDITIONAL TERMS ARE. . . .*)                              3X300490
      IF(M.EQ.0) GO TO 8                                                   3X300500
      DO 7 I=1,M                                                           3X300510
      I1=IJK(2,I)-1                                                        3X300520
      I2=IJK(3,I)-1                                                        3X300530
    7 WRITE(6,887) IJK(1,I),SYM(I1),SYM(I2)                                3X300540
  887 FORMAT(2H (,I2,3H)   ,2A2)                                           3X300550
    8 KU=KASE+1                                                            3X300560
      DO 1000 KKK=1,KU                                                     3X300570
      DO 10 I=1,28                                                         3X300580
      DO 10 J=1,29                                                         3X300590
   10 SUM(I,J)=0.0                                                         3X300600
      N=0                                                                  3X300610
      DO 20 JY=1,36                                                        3X300620
      READ(5,500) J,ANY,V                                                  3X300630
  500 FORMAT(I2,7F10.0)                                                    3X300640
      IF((J.LE.9).OR.(KKK.EQ.2)) GO TO 19                                  3X300650
      DO 15 I=2,5                                                          3X300660
   15 XX(J,I)=V(I-1)                                                       3X300670
      DO 16 I=6,7                                                          3X300680
   16 XX(J,I)=XX(J,I-2)**2                                                 3X300690
      XX(J,8)=XX(J,4)* XX(J,5)                                             3X300700
      DO 18 I=9,10                                                         3X300710
   18 XX(J,I)=XX(J,I-7)**2                                                 3X300720
      DO101 I=11,12                                                        3X300730
  101 XX(J,I)=XX(J,2)*XX(J,I-7)                                            3X300740
      DO 102 I=13,14                                                       3X300750
  102 XX(J,I)=XX(J,3)*XX(J,I-9)                                            3X300760
      IF(M.EQ.0) GO TO 19                                                  3X300770
      DO 1 I=1,M                                                           3X300780
      I0=IJK(1,I)                                                          3X300790
      I1=IJK(2,I)                                                          3X300800
      I2=IJK(3,I)                                                          3X300810
    1 XX(J,I0)=XX(J,I1)*XX(J,I2)                                           3X300820
   19 IF(J.EQ.0) GO TO 25                                                  3X300830
      N=N+1                                                                3X300840
      XX(J,IORS)=ANY                                                       3X300850
      IY(N)=J                                                              3X300860
```

```
      DO 20 K=1,IOR
      DO 20 L=1,IORS
   20 SUM(K,L)=SUM(K,L)+XX(J,K)*XX(J,L)
   25 IF(IPRINT.NE.2) GO TO 26
      DO 1239 JK=1,N
 1239 WRITE(6,789)(XX(JK,I),I=1,14),XX(JK,IORS)
  789 FORMAT(14F4.0,G13.6)
   26 IF(N.LT.IOR) GO TO 60
      WRITE(6,607)
      IF(KKK.EQ.1) WRITE(6,801)
      IF(KKK.EQ.2) WRITE(6,802)
  801 FORMAT(1H+,T17,11H*WAVE DRAG*)
  802 FORMAT(1H+,T17,8H*VOLUME*)
  607 FORMAT(15H0INPUT DATA....)
      DO 55 I=1,N
      NN=IY(I)
      IF(IPRINT.GT.0) WRITE(6,609) NN,XX(NN,IORS)
   55 CONTINUE
  609 FORMAT (I10,G15.7)
  602 FORMAT(10G12.4)
      IF(KKK.EQ.1) CALL CROUT(SUM,B,C,IOR)
      IF(KKK.EQ.2) CALL CROUT(SUM,B,CV,IOR)
      IF((IPRINT.GT.0).AND.(KKK.EQ.1))WRITE(6,803)
  803 FORMAT(T20,*COEFFICIENTS OF WAVE DRAG EQUATION*)
      IF((IPRINT.GT.0).AND.(KKK.EQ.1)) WRITE (6,601) (I,C(I),I=1,IOR)
      IF((IPRINT.GT.0).AND.(KKK.EQ.2)) WRITE(6,601)(I,CV(I),I=1,IOR)
      DO 30 J=1,N
   30 YF(J)=0.0
      DO 40 I=1,N
      DO 40 K=1,IOR
      IF(KKK.EQ.1) YF(I)=YF(I)+XX(I,K)*C(K)
      IF(KKK.EQ.2) YF(I)=YF(I)+XX(I,K)*CV(K)
   40 CONTINUE
C-----------------------------------------------------------------------1
      SUMS=0.
      SUMSUM=0.
      DO 45 K=1,N
      NN=IY(K)
      DEV(NN)=YF(NN)-XX(NN,IORS)
      SUMSUM=SUMSUM+XX(NN,IORS)
   45 SUMS=SUMS+(DEV(NN))**2
      XNY=N
      STD= SQRT(SUMS/XNY)
```

```
      AVG=SUMSUM/XNY                                                    3X301300
      IF(IPRINT.EQ.0) GO TO 1000                                        3X301310
      WRITE(6,699) STD                                                  3X301320
  699 FORMAT(9H STD DEV=,G12.5)                                         3X301330
      IF((IPRINT.GT.0).AND.(KKK.EQ.1)) WRITE(6,804)                     3X301340
  804 FORMAT(T20,*FITTED WAVE DRAG*)                                    3X301350
      WRITE(6,601) (I,YF(I),I=1,N)                                      3X301360
      WRITE(6,710)                                                      3X301370
      WRITE(6,805)                                                      3X301380
  805 FORMAT(T20,*ERROR OF FIT*)                                        3X301390
      WRITE(6,601)(I,DEV(I),I=1,N)                                      3X301400
      WRITE(6,1234)                                                     3X301410
 1234 FORMAT(* +++++++++++++++++++++++++++++++++++++++*)                3X301420
 1000 CONTINUE                                                          3X301430
C----------------------------------------------------------------------L 3X301440
      WRITE(6,1234)                                                     3X301450
  601 FORMAT(5(1X,1H(,I2,1H),G15.7,1X))                                 3X301460
   50 WRITE(6,603)                                                      3X301470
  603 FORMAT(* INPUT NO. OF PTS.*)                                      3X301480
      READ(5,401) (NP(I),I=2,4),NP(1)                                   3X301490
      IF(EOF(5)) 1122,59                                                3X301500
   59 WRITE(6,401) (NP(I),I=2,4),NP(1)                                  3X301510
      WRITE(6,604)                                                      3X301520
  604 FORMAT(* INPUT LOWER LIMITS*)                                     3X301530
      READ(5,403) (O(I),I=2,4),O(1)                                     3X301540
  403 FORMAT(8G9.2)                                                     3X301550
      WRITE(6,403) (O(I),I=2,4),O(1)                                    3X301560
      WRITE(6,605)                                                      3X301570
  605 FORMAT(* INPUT UPPER LIMITS*)                                     3X301580
      READ(5,403) (E(I),I=2,4),E(1)                                     3X301590
      WRITE(6,403) (E(I),I=2,4),E(1)                                    3X301600
C-----------------------------------------------------------------------)3X301610
      CALL SEARCH(O,E,NP,C,IC,NO,YSUM,XMIN,XMAX,YMIN,YMAX,YAVG,IOR,IJK,M3X301620
     & )                                                                3X301630
C-----------------------------------------------------------------------)3X301640
  711 FORMAT(/T7,2HCD,T22,6(A2,13X))                                    3X301650
      WRITE(6,610) NO,IC,YAVG                                           3X301660
  610 FORMAT(I10,16H POINTS REJECTED/I10,16H POINTS ACCEPTED/           3X301670
     + T12,8HAVERAGE=,G12.5)                                            3X301680
      WRITE(6,806)                                                      3X301690
  806  FORMAT(1H0,T10,*MINIMUM DRAG CONFIGURATION*)                     3X301700
      WRITE(6,711) (SYM(I),I=2,4),SYM(1)                                3X301710
      WRITE(6,606) YMIN,(XMIN(I),I=2,4),XMIN(1)                         3X301720
```

```
  710 FORMAT(1H )                                                    3X301730
   77 IF(KASE.NE.1) GO TO 50                                         3X301740
      VMIN=VALUE(XMIN,CV,IOR,IJK,M)                                  3X301750
      WRITE(6,710)                                                   3X301760
      WRITE(6,606) VOLB,VMIN                                         3X301770
      GO TO 50                                                       3X301780
   60 WRITE(6,608) N                                                 3X301790
  608 FORMAT(* INSUFFICIENT NO. OF INPUT DATA*/* N=*,I2)             3X301800
  606 FORMAT(8G15.7)                                                 3X301810
 1122 STOP                                                           3X301820
      END                                                            3X301830
```

```
C******************************************************************XXGEN3X301840
      SUBROUTINE XXGEN(IJK,M)                                           3X301850
      COMMON /XXX/XX(36,29)                                             3X301860
      DIMENSION IJK(3,11)                                               3X301870
      DIMENSION X(45)                                                   3X301880
      DATA X/9*1.,3*-1.,3*0.,3*1.,-1.,0.,1.,-1.,0.,1.,-1.,0.,1.,        3X301890
     & 1.,-1.,0.,-1.,0.,1.,0.,1.,-1.,0.,1.,-1.,-1.,0.,1.,1.,-1.,0./     3X301900
      IC=0                                                              3X301910
      DO 10 J=1,5                                                       3X301920
      DO 10 I=1,9                                                       3X301930
      IC=IC+1                                                           3X301940
      XX(I,J)=X(IC)                                                     3X301950
   10 CONTINUE                                                          3X301960
      DO 15 I=1,36                                                      3X301970
   15 XX(I,1)=1.0                                                       3X301980
      DO 20 J=6,7                                                       3X301990
      DO 20 I=1,9                                                       3X302000
   20 XX(I,J)=XX(I,J-2)**2                                              3X302010
      DO 30 I=1,9                                                       3X302020
   30 XX(I,8)=XX(I,4)*XX(I,5)                                           3X302030
      DO 40 J=9,10                                                      3X302040
      DO 40 I=1,9                                                       3X302050
   40 XX(I,J)=XX(I,J-7)**2                                              3X302060
      DO 50 J=11,12                                                     3X302070
      DO 50 I=1,9                                                       3X302080
   50 XX(I,J)=XX(I,2)*XX(I,J-7)                                         3X302090
      DO 60 J=13,14                                                     3X302100
      DO 60 I=1,9                                                       3X302110
   60 XX(I,J)=XX(I,3)*XX(I,J-9)                                         3X302120
      IF(M.EQ.0) RETURN                                                 3X302130
      DO 80 I=1,9                                                       3X302140
      DO 80 J=1,M                                                       3X302150
      I0=IJK(1,J)                                                       3X302160
      I1=IJK(2,J)                                                       3X302170
      I2=IJK(3,J)                                                       3X302180
      XX(I,I0)=XX(I,I1)*XX(I,I2)                                        3X302190
   80 CONTINUE                                                          3X302200
      RETURN                                                            3X302210
      END                                                               3X302220
```

```
C*******************************************************************CROUT
      SUBROUTINE CROUT(A,AUX,B,N)
      DIMENSION A(28,29),AUX(28,29),B(1)
C     A IS GIVEN MATRIX, AUX IS AUXILARY MATRIX, B IS SOLUTION VECTOR
      NN = N+1
      DO 1 J=1,N
      DO 1 K=1,NN
    1 AUX(J,K) = A(J,K)
      DO 2 J=2,NN
    2 AUX(1,J) = A(1,J)/AUX(1,1)
      DO 5 J=2,N
      K = J-1
      DO 5 JJ=J,N
      JJP1 = JJ+1
      DO 4 L=1,K
      AUX(JJ,J) = AUX(JJ,J) - AUX(L,J)*AUX(JJ,L)
    4 AUX(J,JJP1) = AUX(J,JJP1) - AUX(L,JJP1)*AUX(J,L)
    5 AUX(J,JJP1) = AUX(J,JJP1)/AUX(J,J)
      DO 6 J=1,N
    6 B(J) = AUX(J,NN)
      DO 7 J=2,N
      K = NN-J
      M = J-1
      DO 7 L=1,M
      NNL = NN-L
    7 B(K) = B(K) - AUX(K,NNL)*B(NNL)
      RETURN
      END
```

```
C*****************************************************************SEARCH3X302510
      SUBROUTINE SEARCH(O,E,NP,C,IC,NO,SUM,XMIN,XMAX,YMIN,YMAX,YAVG,IOR,3X302520
     & IJK,M)                                                           3X302530
      COMMON/CHECK/FS(20),ZZB(8,25),CV(28),VOLB,KIND(20),NS             3X302540
      DIMENSION O(4),E(4),NP(4),D(4),X(4),C(1),XMAX(4),XMIN(4),IJK(3,11)3X302550
      DO 10 I=1,4                                                       3X302560
      IF((NP(I).LE.0))NP(I)=1                                           3X302570
      XN=NP(I)-1                                                        3X302580
      D(I)=0.                                                           3X302590
      IF(XN.NE.0.) D(I)=(E(I)-O(I))/XN                                  3X302600
   10 CONTINUE                                                          3X302610
      SUM=0.                                                            3X302620
      NO=0                                                              3X302630
      IC=0                                                              3X302640
      YMAX=-1.0E50                                                      3X302650
      YMIN=1.0E50                                                       3X302660
      N1=NP(1)                                                          3X302670
      DO 1000 I1=1,N1                                                   3X302680
      XI1=I1-1                                                          3X302690
      X(1)=O(1)+XI1*D(1)                                                3X302700
      N2=NP(2)                                                          3X302710
      DO 1000 I2=1,N2                                                   3X302720
      XI2=I2-1                                                          3X302730
      X(2)=O(2)+XI2*D(2)                                                3X302740
      N3=NP(3)                                                          3X302750
      DO 1000 I3=1,N3                                                   3X302760
      XI3=I3-1                                                          3X302770
      X(3)=O(3)+XI3*D(3)                                                3X302780
      N4=NP(4)                                                          3X302790
      DO 1000 I4=1,N4                                                   3X302800
      XI4=I4-1                                                          3X302810
      X(4)=O(4)+XI4*D(4)                                                3X302820
      N5=NP(5)                                                          3X302830
      VOL=VALUE(X,CV,IOR,IJK,M)                                         3X302840
      IF(VOL.GE.VOLB) GO TO 50                                          3X302850
      NO=NO+1                                                           3X302860
      GO TO 1000                                                        3X302870
   50 Y=VALUE(X,C,IOR,IJK,M)                                            3X302880
      IC=IC+1                                                           3X302890
      SUM=SUM+Y                                                         3X302900
      IF (Y.LT.YMAX) GO TO 200                                          3X302910
      DO 100 I=1,4                                                      3X302920
  100 XMAX(I)=X(I)                                                      3X302930
```

```
      YMAX=Y                                                    3X302940
  200 IF(Y.GT.YMIN) GO TO 1000                                  3X302950
      DO 300 I=1,4                                              3X302960
  300 XMIN(I)=X(I)                                              3X302970
      YMIN=Y                                                    3X302980
 1000 CONTINUE                                                  3X302990
      XIC=IC                                                    3X303000
      YAVG=SUM/XIC                                              3X303010
      RETURN                                                    3X303020
      END                                                       3X303030
```

```
C******************************************************************VALUE3X303040
      FUNCTION VALUE(X,C,IOR,IJK,M)                                     3X303050
      DIMENSION X(1),C(1),Y(36),IJK(3,11)                               3X303060
      VALUE=C(1)                                                        3X303070
      DO 10 I=2,5                                                       3X303080
   10 Y(I)=X(I-1)                                                       3X303090
      DO 20 I=6,7                                                       3X303100
   20 Y(I)=X(I-3)**2                                                    3X303110
      DO 30 I=8,8                                                       3X303120
   30 Y(I)=X(3)*X(4)                                                    3X303130
      DO 40 I=9,10                                                      3X303140
   40 Y(I)=X(I-8)**2                                                    3X303150
      IF(M.EQ.0) GO TO 60                                               3X303160
      DO 50 I=1,M                                                       3X303170
      I0=IJK(1,I)                                                       3X303180
      I1=IJK(2,I)                                                       3X303190
      I2=IJK(3,I)                                                       3X303200
   50 Y(I0)=Y(I1)*Y(I2)                                                 3X303210
   60 DO 70 I=2,IOR                                                     3X303220
   70 VALUE=VALUE+C(I)*Y(I)                                             3X303230
      RETURN                                                            3X303240
      END                                                               3X303250
```

```
C***FUSELAGE VOLUME CALCULATION PROCEDURE
      DIMENSION ZX(13),P(13),Q(13),R(13),S(13),T(13),SG(13),ZXP(13)
      DIMENSION AREA(101),YS(101)
      REWIND 2
      NY=101
      M78=10
      YMAXL=-100.
      YMAX=-100.
   10 WRITE(6,600)
  600 FORMAT(' INPUT KASE,Y1,Y2')
      READ(K,501) KASE,Y1,Y2
  501 FORMAT(I5,2F10.2)
      XN=NY-1
      DELTAY=(Y2-Y1)/XN
      DO 50 LL=1,NY
      XLL=LL-1
      Y=Y1+DELTAY*XLL
       YS(LL)=Y
      DO 2 I=1,50
      IF((Y.LE.YMAX).AND.(Y.GE.YMAXL)) GO TO 3
      IF(Y.LT.YMAXL) REWIND 2
      YMAXL=YMAX
      READ(2,200) YMAX,N
      M=7
      DO 2 K=1,N
      READ(2,201) L,P(L),Q(L),R(L),S(L),T(L),SG(L)
      IF(L.GT.7) M=13
      IF((L.EQ.9).OR.(L.EQ.8)) M78=8
    2 CONTINUE
  200 FORMAT(F10.5,I5)
  201 FORMAT(I2,5E15.8,F2.0)
    3 DO 4 I=1,7
      CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
    4 CONTINUE
      IF(M.EQ.7) GO TO 6
      DO 5 I=M78,13
    5 CALL CONICF(P(I),Q(I),R(I),S(I),T(I),SG(I),Y,ZX(I),ZXP(I))
      IF(M78.EQ.10) SG(8) = 99.
    6 CALL PLOT(Y,ZX,ZXP,M,SG,AREA(LL))
      MODE=1
   50 CONTINUE
      CALL SIGMA2(NY,0,YS,AREA,VOL)
      WRITE(6,700) KASE,VOL
```

```
700 FORMAT('1KASE =',I5,'VOLUME =',G14.6)
    READ(5,500) DUMMY
500 FORMAT(F20.0)
    GO TO 10
    END
```

```
      SUBROUTINE CONICF(P,Q,R,S,T,SG,X,Y,YP)
      ROOT=SG*SQRT((R*X+S)*X+T)
      Y=P*X+Q+ROOT
      IF(SG.NE.0.) GO TO 10
      YP=P
      GO TO 20
   10 YP=P+(R*X+S/2.0)/ROOT
   20 RETURN
      END
```

```
      SUBROUTINE PLOT(Y,ZX,ZXP,M,SG,AREA)
      COMMON PRINT,NMAX
      DIMENSION XP(200),ZP(200)
      DIMENSION ZX(1),ZXP(1),SG(1)
      N=200
      XN=N-1
      NM1=N-1
      IF(M.EQ.7) Z1=ZX(1)
      IF(M.EQ.13) Z1=ZX(10)
      Z2=ZX(3)
      XP(1)=0.
      XP(N)=0.
      ZP(1)=Z1
      ZP(N)=Z2
      DZ=(Z1-Z2)/XN
      DO 1000 I=2,NM1
      XI=I-1
      Z=Z1-DZ*XI
      ZP(I)=-Z
      IF((M.EQ.7).OR.(SG(8).EQ.99.)) GO TO 10
      IF(Z.GT.ZX(8)) GO TO 100
      IF(Z.GT.ZX(9)) GO TO 200
   10 IF(Z.GT.ZX(2)) GO TO 300
      IF(Z.GT.ZX(3)) GO TO 400
C***LOWER FLAT
      INDEX=5
      X=ZX(6)
      GO TO 500
C***CANOPY
  100 INDEX=1
      X=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),Z)+ZX(13)
      GO TO 500
C***CUBIC
  200 INDEX=2
      IF(SG(8).NE.99.) GO TO 15
      IF(Z.GE.ZX(1)) GO TO 300
      GO TO 400
   15 X1=XELIPS(ZX(11),ZX(10),ZX(12)-ZX(13),ZX(8))+ZX(13)
      X2=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),ZX(9))+ZX(4)
      Z1P=XELIPP(ZX(11),ZX(10),X1,ZX(12),ZX(8))
      Z2P=XELIPP(ZX(2),ZX(1),X2,ZX(5)-ZX(4),ZX(9))
      DZ1=Z-ZX(8)
      DZ21=ZX(9)-ZX(8)
```

```
      A=3.*DZ21-(Z2P+2.*Z1P)*(X2-X1)
      B=-2.*DZ21+(Z2P+Z1P)*(X2-X1)
      IF((Z1P.LT.-4.).OR.(Z2P.LT.-4.).OR.(Z1P.GE.0.)) GO TO 22
      GO TO 18
   22 TX=1
      IF(Z.EQ.ZX(8)) X=X1
      IF(Z.EQ.ZX(9)) X=X2
      IF((Z.EQ.ZX(8)).OR.(Z.EQ.ZX(9))) GO TO 500
      XI=X1
      DO 19 L=1,2
      DO 17 J=1,6
      DXR=(XI-X1)/(X2-X1)
      ZI=ZX(8)+Z1P*(XI-X1)+A*DXR**2+B*DXR**3
      ZIP=Z1P+(2.0*A*DXR+3.0*B*DXR**2)/(X2-X1)
      IF(ZIP.NE.0.) GO TO 16
      X=XI+(X2-X1)/100.
      GO TO 17
   16 X=XI-(ZI-Z)/ZIP
      IF(ABS(X-XI).LE.1.E-5) GO TO 500
   17 XI=X
      XI=X2
   19 CONTINUE
      GO TO 500
   18 DX21=X2-X1
      X1P=1.0/Z1P
      X2P=1.0/Z2P
      A=3.0*DX21-(X2P+2.0*X1P)*DZ21
      B=-2.0*DX21+(X2P+X1P)*DZ21
      DZR=(Z-ZX(8))/DZ21
      X=X1+X1P*DZ1+A*DZR**2+B*DZR**3
      GO TO 500
C***UPPER ELLIPSE
  300 INDEX=3
      X=XELIPS(ZX(2),ZX(1),ZX(5)-ZX(4),Z)  + ZX(4)
      GO TO 500
C***LOWER CONIC
  400 INDEX=4
      CALL CONICS(ZX,ZXP,6,3,5,2,7,Z,X)
  500 XP(I)=X
 1000 CONTINUE
      CALL SIGMA2(N,0,ZP,XP,AREA)
      RETURN
      END
```

```
      FUNCTION XELIPS(Z1,Z2,X,Z)
      XELIPS=X*SQRT(1.0-((Z-Z1)/(Z2-Z1))**2)
      XELIPS=ABS(XELIPS)
      RETURN
      END
```

```
FUNCTION XELIPP(Z1,Z2,X1,X,Z)
XELIPP=-X1/(Z-Z1)*((Z2-Z1)/X)**2
RETURN
END
```

```
SUBROUTINE CONICS(ZX,ZXP,ICX,ICZ,ISX,ISZ,IB,Z,X)
DIMENSION ZX(1),ZXP(1)
SIG = SIGN(1.,ZX(ICZ)-ZX(ISZ))
H = SIG*(1.414213562*ZX(IB)-ZX(ISX))-ZX(ISZ)
ZZ = Z-ZX(ISZ)
Z1 = ZX(ICZ)-ZX(ISZ)
X1 = ZX(ICX)-ZX(ISX)
CK = (H-Z1)**2/(4.*H*X1*Z1*(H-SIG*X1-Z1))
A = 2.*X1*Z1*ZZ*CK+Z1-ZZ
X = ZX(ISX)+(A-SIG*SQRT((2.*A-Z1+ZZ)*(Z1-ZZ)))/(2.*Z1*Z1*CK)
RETURN
END
```

```
      SUBROUTINE SIGMA2(N,IND,X,Y,SUM)
      DIMENSION X(1),Y(1),SUM(1)
      IS = 1
      SUM(1) = 0.
      DO 40 I=2,N
      ISP = IS+IND
      AREA = (Y(I)+Y(I-1))*(X(I)-X(I-1))/2.0
      SUM(ISP) = SUM(IS)+AREA
   40 IS = ISP
      RETURN
      END
```

SECTION VI

FUNCTIONAL DESCRIPTION OF SUBROUTINES

XYGEN	-	Generates the Latin Square Coordinate Matrix.
CROUT	-	Uses Crout method to solve simultaneous linear equations.
SEARCH	-	Searches for the minimum value.
VALUE	-	Uses the Latin Square Surface Fit Wave Drag Coefficients.
XFORM	-	Transforms reduced variables to physical variables for forward fuselage.
SIGMA2	-	Performs integration using the second order Lagrange formula.
CNSTN	-	Imposes constraints according to input parameter.
BASEA	-	A second entry of CNSTN to transfer the value of the baseline cross-sectional area.
VOLMD	-	A third entry of CNSTN to transfer the value of the fuselage volume.
AREA	-	Computes the cross-sectional area of fuselage excluding the canopy.
KONIC	-	Computes fuselage widths at given fuselage stations.
KONICA	-	A second entry of KNOIC to compute the cross-sectional area of the lower fuselage.
TABLE	-	Compares the fuselage width distribution with given local tabulation of width or with randome contour.
READ	-	Reads baseline body description data.
CONICF	-	Computes Y and Y' from given conic coefficients.
LINEAR	-	Linear interpolation routine.

Appendix

SAMPLE DECK SET-UP

Samples of deck set-up are given below. The IVS and 3DMoC programs, because of their lengths, are stored on tapes in an updatable binary form. Permanent files of these programs are created for computation. The Surface Fit and Minimum Search Programs are read in the computer to create permanent files for computation. The samples were selected to illustrate various calculations for wave drag reduction of an F-4 type baseline configuration.

(1) LOAD IVS AND 3DMOC PROGRAMS ON TAPE, CREATE PF TO RUN FUSELAGE-WINGS FOR 100 SURFACES, AND STORE OUTPUT DATA ON DATA TAPE.

```
JOB1,CB100000,T500,NT2.
*ACCOUNT CARD*
RESOURC(NT=2)
REQUEST(A,D=HD,VSN=4379),NT,PO=W)
UPDATE(N,W)
REWIND(NEWPL)
COPYBF(NEWPL,A)
FTN(I=COMPILE)
SAVE(LGO=IVSPF)
UPDATE(N=NUPL,C=COMT,W)
REWIND(NUPL)
COPYBF(NUPL,A)
FTN(I=COMT,B=BIN)
SAVE(BIN=MOCPF2)
SETCORE(0)
LOAD(LGO)
BLUNT.
XPAUSE.  **** IF TAPE 4380 REACH END, USE TAPE 4381 AS NEXT OUTPUT TAPE ****
VSN(TAPE1=4380/4381)
REQUEST(TAPE1,D=HD,VSN=4380,NT,PO=W)
SETCORE(0)
REDUCE.
BIN.
7/8/9
*IVS PROGRAM
7/8/9
*3DMOC2 PROGRAM
7/8/9
  $$$$$$$$$$$ F-4 BASELINE WITH WINGS $$$$$$$$$$$$$$$$$$$
    2.5          1.4          0.          35.70         5.          1.05
    1.            .001        4.           0.          90.          6.
1                 .1
7/8/9
  $$$$$$$$$$$ F-4 BASELINE WITH WINGS $$$$$$$$$$$$$$$$$$$
    .1          .1          .1          .01        0.       430.        3.       110.
  100     5   11    14    1     0    20
  45.         11     0    0
  65.         17    15   17
   85.        21    13   21
  110.        22    13   19     5     9
```

```
 115.        22   15   19    4    6    6    6    8    1
 120.        24   17   21    4    6    6    6    9    2
 140.        26   19   23    4    6    6    6   10    3
 165.        36   29   33    4    5   10   10   15    4
 185.        38   31   35    4    6   10   11   15    3
 205.        43   36   40    4    6   11   13   18    4
 250.        44   40   44    3    5   13   16   20    3
 290.        46   42   46    3    5   13   17   21    3
 330.        48   44   48    3    5   13   18   22    3
 370.        50   46   50    3    5   13   19   23    3
   2.5      1.4        0.0
 -26.105   11.015      1.       530.          0.5
* FUSELAGE + WINGS BODY DESCRIPTION DATA *
6/7/8/9
```

(2) LOAD PHASE-2 3DMOC PROGRAM ON TAPE AND CREATE PF.

```
JOB2,CB100000,T500.
* ACCOUNT CARD *
UPDATE(N,W)
FTN(I=COMPILE)
SAVE(LGO=MOCPF1)
7/8/9
*3DMOC1 PROGRAM
6/7/8/9
```

(3) LOAD LATEIN1 PROGRAM TO CREATE PF AND RUN RADOME CONSTRAINT OPTION.

```
JOB3,CB600000,T500.
* ACCOUNT CARD *
FTN.
SAVE(LGO=5X5)
COPYBR(INPUT,TAPE3)
REWIND(TAPE3)
LGO.
7/8/9
*LATIN1 LATIN SQUARE MINIMUM VALUE SEARCH PROGRAM *
7/8/9
*FUSELAGE BODY DESCRIPTION DATA*
7/8/9
    L     B     F     H    A'     A
   23     1
    3
  50.       19.      18.5
* BLANK CARD *
 1 8.72259                                                      00000080
 2 5.97017                                                      00000090
 3 5.72492                                                      00000100
 4 5.88469                                                      00000110
 5 5.95256                                                      00000120
 6 7.62580                                                      00000130
 7 7.25404                                                      00000140
 8 7.10510                                                      00000150
 9 5.43938                                                      00000160
10 5.76062                                                      00000170
11 6.52684                                                      00000180
12 6.67305                                                      00000190
13 6.59154                                                      00000200
14 6.86143                                                      00000210
15 7.61059                                                      00000220
16 8.21773                                                      00000230
17 8.68830                                                      00000240
18 6.43817                                                      00000250
19 6.61810                                                      00000260
20 5.93960                                                      00000270
21 7.63046                                                      00000280
22 8.22275                                                      00000290
23 8.62522                                                      00000300
24 7.24846                                                      00000310
```

```
25 5.87617                                                                  00000330
26 6.69437   -1.38     0.        .6666667 -.6666667  .6666667  -.4         00000340
27 6.57260   -1.38     1.5       2.0      -.6666667  2.0       -2.0        00000350
28 6.09569   -1.38     2.0       2.0        -2.0     2.0       -2.0        00000360
29 5.95529   -1.38     0.        2.0        -2.0     2.0       -2.0        00000370
30 5.86343   -1.38     -1.71     -2.0       -2.0     .84       -2.0        00000380
31 6.81449   -1.38     2.0       2.0        -2.0     1.91      .5325       00000390
32 6.11884   -0.38     2.0       2.0        -2.0     1.8       -.418       00000400
* BLANK CARD *
    1    5    5    5    5    5
   0.       -2.       -2.       -2.       -2.        -2
  0.        2.         2.         2.       2.         2.
6/7/8/9
```

(4) RESTART FUSELAGE-WING PROGRAM TO CONTINUE FOR 10 MORE SURFACES.

```
DER1,CB60000,T500,TP1.
* ACCOUNT CARD *
VSN(TAPE1=4380/4381)
XPAUSE.  **** IF TAPE 4380 REACH END. USE TAPE 4381 AS NEXT OUTPUT TAPE ****
REQUEST(A,D=HD,VSN=4379),NT,PO=W)
SKIPF(TAPE1,99,17,B)
GET(BIN=MOCPF2)
SETCORE(0)
REDUCE.
BIN.
7/8/9
  $$$$$$$$$$$ F-4 BASELINE WITH WING $$$$$$S $$$$$$$$$$$$
    .1         .1          .1          .01       0.       430.        3.       110.
  110     5    11    14     1   100    20
   45.         11     0     0
   65.         17    15    17
    85.        21    13    21
   110.        22    13    19     5     9
 115.          22    15    19     4     6     6     6     8     1
 120.          24    17    21     4     6     6     6     9     2
 140.          26    19    23     4     6     6     6    10     3
 165.          36    29    33     4     5    10    10    15     4
 185.          38    31    35     4     6    10    11    15     3
 205.          43    36    40     4     6    11    13    18     4
 250.          44    40    44     3     5    13    16    20     3
 290.          46    42    46     3     5    13    17    21     3
 330.          48    44    48     3     5    13    18    22     3
 370.          50    46    50     3     5    13    19    23     3
   2.5         1.4         0.0
  -26.105     11.015       1.        530.           0.5
* FUSELAGE + WINGS BODY DESCRIPTION DATA *
6/7/8/9
```

(5) USE PF TO CALCULATE FUSELAGE FLOW FIELD.

```
DER2,CM100000,T500.
* ACCOUNT CARD *
GET(BIN=IVSPF)
SETCORE(0)
LOAD(BIN)
BLUNT.
GET(LGO=MOCPF1)
SETCORE(0)
REDUCE.
LGO.
7/8/9
  $$$$$$$$$$$ F-4 BASELINE FUSELAGE ONLY $$$$$$$$$$$$$$$$
    2.5          1.4          0.          35.70        5.          1.05
    1.            .001        4.           0.          90.          6.
1                 .1
7/8/9
  $$$$$$$$$$$ F-4 BASELINE FUSELAGE ONLY $$$$$$$$$$$$$$$$
    .01          .01          .01          .01         0.         230.
   10     5    11     7     1     0     5
  45.          11     0     0
  65.          17    15    17
  85.          21    13    21
  95.          21    12    20
 145.          21    11    19
 190.          21    11    21
 240.          15     0     0
   2.5          1.4         0.0
 -26.105      11.015        1.        530.
*FUSELAGE BODY DESCRIPTION DATA*
6/7/8/9
```

(6) USE LATIN1 PF TO RUN EQUIPMENT CONSTRAINT OPTION

```
DER3,CB60000,T500.
* ACCOUNT CARD *
COPYBR(INPUT,TAPE3)
REWIND(TAPE3)
GET(LGO=5X5)
LGO.
7/8/9
*FUSELAGE BODY DESCRIPTION DATA*
7/8/9
    L      B     F     H    A'     A
   23      1
    2
  50.       0.
* BLANK CARD *
 1 8.72259                                                                                          00000080
 2 5.97017                                                                                          00000090
 3 5.72492                                                                                          00000100
 4 5.88469                                                                                          00000110
 5 5.95256                                                                                          00000120
 6 7.62580                                                                                          00000130
 7 7.25404                                                                                          00000140
 8 7.10510                                                                                          00000150
 9 5.43938                                                                                          00000160
10 5.76062                                                                                          00000170
11 6.52684                                                                                          00000180
12 6.67305                                                                                          00000190
13 6.59154                                                                                          00000200
14 6.86143                                                                                          00000210
15 7.61059                                                                                          00000220
16 8.21773                                                                                          00000230
17 8.68830                                                                                          00000240
18 6.43817                                                                                          00000250
19 6.61810                                                                                          00000260
20 5.93960                                                                                          00000270
21 7.63046                                                                                          00000280
22 8.22275                                                                                          00000290
23 8.62522                                                                                          00000300
24 7.24846                                                                                          00000310
25 5.87617                                                                                          00000330
26 6.69437   -1.38     0.        .6666667 -.6666667  .6666667  -.4     00000340
27 6.57260   -1.38     1.5       2.0      -.6666667  2.0       -2.0    00000350
```

```
28 6.09569   -1.38     2.0      2.0      -2.0     2.0      -2.0     00000360
29 5.95529   -1.38     0.       2.0      -2.0     2.0      -2.0     00000370
30 5.86343   -1.38     -1.71    -2.0     -2.0     .84      -2.0     00000380
31 6.81449   -1.38     2.0      2.0      -2.0     1.91     .5325    00000390
32 6.11884   -0.38     2.0      2.0      -2.0     1.8      -.418    00000400
* BLANK CARD *
   1    5    5    5    5    5
   0.      -2.     -2.      -2.      -2.      -2
  0.      2.       2.       2.       2.       2.
6/7/8/9
```

(7) USE LATIN1 PROGRAM TO RUN CROSS-SECTIONAL AREA CONSTRAINT.

```
CHU1,CB60000,T500.
* ACCOUNT CARD *
COPYBR(INPUT,TAPE3)
REWIND(TAPE3)
GET(LGO=5X5)
LGO.
7/8/9
*FUSELAGE BODY DESCRIPTION DATA*
7/8/9
    L    B    F    H   A'    A
   23    1
    1
  50.       0.95
* BLANK CARD *
 1 8.72259
 2 5.97017
 3 5.72492
 4 5.88469
 5 5.95256
 6 7.62580
 7 7.25404
 8 7.10510
 9 5.43938
10 5.76062
11 6.52684
12 6.67305
13 6.59154
14 6.86143
15 7.61059
16 8.21773
17 8.68830
18 6.43817
19 6.61810
20 5.93960
21 7.63046
22 8.22275
23 8.62522
24 7.24846
25 5.87617
26 6.69437   -1.38     0.        .6666667 -.6666667  .6666667  -.4       00000340
27 6.57260   -1.38     1.5       2.0      -.6666667  2.0       -2.0      00000350
```

```
28 6.09569   -1.38      2.0        2.0        -2.0       2.0        -2.0       00000360
29 5.95529   -1.38      0.         2.0        -2.0       2.0        -2.0       00000370
30 5.86343   -1.38      -1.71      -2.0       -2.0       .84        -2.0       00000380
31 6.81449   -1.38      2.0        2.0        -2.0       1.91       .5325      00000390
32 6.11884   -0.38      2.0        2.0        -2.0       1.8        -.418      00000400
* BLANK CARD *
    1    5    5    5    5    5
   0.       -2.       -2.       -2.       -2.        -2
  0.        2.         2.         2.       2.         2.
6/7/8/9
```

(8) USE LATIN1 PROGRAM TO RUN VOLUME CONSTRAINT OPTION.

```
CHU2,CB60000,T500.
* ACCOUNT CARD *
COPYBR(INPUT,TAPE3)
REWIND(TAPE3)
GET(LGO=5X5)
LGO.
7/8/9
*FUSELAGE BODY DESCRIPTION DATA*
7/8/9
   L     B     F     H    A'     A
  23     1
   4
   0.        230.     1.          11    1
  90.       150.     190.     200.     205.     210.     215.     220.
 225.
* BLANK CARD *
 1 8.72259                                                     00000080
 2 5.97017                                                     00000090
 3 5.72492                                                     00000100
 4 5.88469                                                     00000110
 5 5.95256                                                     00000120
 6 7.62580                                                     00000130
 7 7.25404                                                     00000140
 8 7.10510                                                     00000150
 9 5.43938                                                     00000160
10 5.76062                                                     00000170
11 6.52684                                                     00000180
12 6.67305                                                     00000190
13 6.59154                                                     00000200
14 6.86143                                                     00000210
15 7.61059                                                     00000220
16 8.21773                                                     00000230
17 8.68830                                                     00000240
18 6.43817                                                     00000250
19 6.61810                                                     00000260
20 5.93960                                                     00000270
21 7.63046                                                     00000280
22 8.22275                                                     00000290
23 8.62522                                                     00000300
24 7.24846                                                     00000310
25 5.87617                                                     00000330
```

```
26 6.69437   -1.38     0.        .6666667 -.6666667  .6666667  -.4       00000340
27 6.57260   -1.38     1.5       2.0      -.6666667  2.0       -2.0      00000350
28 6.09569   -1.38     2.0       2.0        -2.0     2.0       -2.0      00000360
29 5.95529   -1.38     0.        2.0        -2.0     2.0       -2.0      00000370
30 5.86343   -1.38     -1.71     -2.0       -2.0     .84       -2.0      00000380
31 6.81449   -1.38     2.0       2.0        -2.0     1.91      .5325     00000390
32 6.11884   -0.38     2.0       2.0        -2.0     1.8       -.418     00000400
* BLANK CARD *
 1 27969.                                                                00000420
 2 24190.                                                                00000430
 3 22492.                                                                00000440
 4 27633.                                                                00000450
 5 31198.                                                                00000460
 6 25953.                                                                00000470
 7 26918.                                                                00000480
 8 31499.                                                                00000490
 9 25273.                                                                00000500
10 30202.                                                                00000510
11 22460.                                                                00000520
12 26212.                                                                00000530
13 29430.                                                                00000540
14 34228.                                                                00000550
15 37811.                                                                00000560
16 28259.                                                                00000570
17 33819.                                                                00000580
18 29035.                                                                00000590
19 32614.                                                                00000600
20 30919.                                                                00000610
21 26603.                                                                00000620
22 31917.                                                                00000630
23 35621.                                                                00000640
24 36351.                                                                00000650
25 31103.                                                                00000660
26 25076.
27 25055.
28 23565.
29 21192.
30 20452.
31 27049.
32 27194.
* BLANK CARD
    1     5     5     5     5     5
   0.       -2.        -2.        -2.         -2.          -2
```

0. 6/7/8/9	2.	2.	2.	2.	2.

(9) LOAD LATINT2 PROGRAM TO CREATE PF AND RUN VOLUME CONSTRAINT OPTION.

```
CHU3,CB60000,T500.
* ACCOUNT CARD *
FTN.
SAVE(LGO=3X3)
LGO.
7/8/9
* LATIN2 LATIN SQUARE AND MINIMUM VALUE SEARCH PROGRAM *
7/8/9
    H    A   R1   R2
   10    1
    1 116486.8
 1 .0104602
 4 .0094555
 7 .0116629
 2 .0116525
 5 .0104013
 8 .0114297
 3 .0107868
 6 .0128320
 9 .0114309
10.00979359 -.333333  -.2       -1.0      -1.0
* BLANK CARD
 1 -70402.3
 4 -48181.6
 7 32094.0
 2 1265.5
 5 36291.1
 8 83446.1
 3 47920.1
 6 137459.1
 9 154575.0
10 0.0
* BLANK CARD
   21   21   21   21
  -1.      -1.       -1.       -1.
   1.       1.        1.        1.
6/7/8/9
```

FUSELAGE BODY DESCRIPTION DATA

```
-26.511737     7
 1 5.49149075E-01 5.73367672E+01 3.22022279E-01 6.24456002E+00 8.89238536E+02-1
 2 0.0          E+00 1.10150000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 3 0.0          E+00 1.10150000E+01-1.0          E+00-5.22100000E+01-6.80471025E+02-1
 4 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
 5 8.14509253E-01 3.20504355E+01 3.09440900E-01 1.81910169E+01 3.55338926E+02-1
 6 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
 7 3.36009329E+00 1.58223695E+02 1.09128778E+01 9.80325854E+02 2.40943954E+04-1
-26.105        1
 3-2.49023891E-01-4.04856541E+01 3.26127292E-01 3.22161029E-02 1.70631329E+03 1
   0.          1
 2 9.41050000E-02 1.34716110E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
  35.          1
 5 2.37961562E-01 1.97192947E+01 4.59075164E-02-1.22612884E+00 4.25012054E+01-1
  50.          1
 3 4.69618114E-02-6.35703498E+00 8.76793612E-03-7.46504489E-01 1.73845541E+01 1
  60.         12
 1 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
 2 9.41050000E-02 1.34716110E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 3 6.17342129E-02-6.76706983E+00 6.00394077E-03-5.32392375E-01 1.27723780E+01 1
 5 2.43645384E-01 1.97395403E+01 4.84059774E-02-1.25659158E+00 4.38354577E+01-1
 6 5.61007015E-01-1.87201403E+01-3.62840714E-01 4.67526985E+01-1.34348031E+03-1
 7 6.65006015E-02 8.65811699E+00 2.55118541E-03-3.82669717E-01 1.95318964E+01-1
 8 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
 9 1.03303975E+00-2.69645401E+02 1.48693730E+00-6.81728634E+02 9.69296900E+04 1
10 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
11 4.60000000E-01-3.00000000E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
12 9.25000000E-02 7.77500000E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
13 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
  65.          5
 1 2.50000000E-01 2.65000000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 3 2.20660324E-01-8.51318827E+00 5.04189058E-02-4.82042026E+00 1.46485599E+02-1
 8 2.50000000E-01 2.65000000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
10 3.30800117E-01 2.13816842E+01 2.10071510E-02-2.56454008E+00 7.83197281E+01 1
13 1.69428355E-01-7.61103902E+00 3.62521514E-02-5.18344403E+00 1.87025196E+02-1
 1 4.49999988E-01 1.12031262E+01 1.83749962E-01-2.48062449E+01 8.41337958E+02 1
 6 7.62467575E-01-5.13706401E+01-6.28039714E-02 5.64312348E+00-5.50767734E+01 1
10 4.65789000E-01 1.32237330E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
  77.          3
 1 3.19469309E-01 2.20894279E+01-3.44198713E-02 5.18085882E+00-1.93702750E+02 1
 5 7.24999081E-02 1.70083294E+01 1.12293182E-03-2.02127204E-01 9.31348979E+00 1
```

```
8 1.81818203E-01 3.21818144E+01 1.03305539E-02-1.57024414E+00 6.01238176E+01-1
 80.        3
 1 2.63562000E-01 2.74657260E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 6 5.37634000E-03 9.58602200E+00 0.0         E+00 0.0         E+00 1.0         E+00 0
10 4.65789000E-01 1.32237330E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 85.        2
 2 5.36480156E-02 1.54704910E+01-2.71605902E-03 5.34714063E-01-2.38625259E+01 1
10 4.65789000E-01 1.32237330E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
100.        2
 8 1.29323524E-01 3.31659571E+01 8.33241401E-03-1.55383488E+00 7.73570984E+01 1
13-4.61624110E-01 5.15383731E+01 2.62234164E-01-5.23353921E+01 2.62443194E+03-1
103.        1
11 2.94177736E-01 2.02578527E+01 5.59112031E-02-1.33961752E+01 8.25069479E+02-1
110.        2
10 3.72282544E-01 2.47202489E+01 1.23135674E-02-2.88544023E+00 1.70045253E+02-1
13 0.0         E+00 0.0         E+00 0.0         E+00 0.0         E+00 1.0         E+00 0
120.        2
 5 4.22221093E-02 2.08296406E+01-7.26924572E-04 1.97586566E-01-1.28325043E+01 1
12 3.11818358E-02 1.37379328E+01-2.04450978E-03 5.45701343E-01-3.46769581E+01 1
122.        2
 2 0.0         E+00 2.30000000E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
10 3.72282544E-01 2.47202489E+01 1.23135674E-02-2.88544023E+00 1.70045253E+02-1
130.        1
10 1.44921931E-01 4.55285164E+01-3.31860250E-02 1.01251470E+01-7.07791365E+02 1
140.        2
 8 2.05479000E-01 2.72877300E+01 0.0         F+00 0.0         E+00 1.0         E+00 0
 9 2.26027000E-01 1.81164900E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
150.        1
11 1.26667000E-01 3.69666200E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
160.        2
 1 3.58105141E-01 2.40204689E+01 5.91180929E-02-1.57053527E+01 1.14091266E+03-1
12-4.35421248E-02 2.43167648E+01-1.46009950E-03 5.87340493E-01-5.23091400E+01 1
170.        2
 5 0.0         E+00 2.80000000E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
10 1.44921931E-01 4.55285164E+01-3.31860250E-02 1.01251470E+01-7.07791365E+02 1
180.        2
 6 3.03462490E-02 4.93794747E+00 3.98463449E-03-1.37491097E+00 1.18741491E+02 1
11 2.35939575E-02 4.64431781E+01-6.09878510F-02 2.23944900E+01-1.97977872E+03 1
190.        3
 3 0.0         E+00 1.00000000E+00 0.0         E+00 0.0         E+00 1.0         E+00 0
 6 8.00000000E-02-3.40000000E+00 0.0         E+00 0.0         E+00 1.0         E+00 0
10-2.48597019E-02 8.23034310E+01-1.04050162E-04 4.10035098E-02-4.00432153E+00 1
203.        2
```

```
 1 8.00000602E-02 6.23787330E+01 1.65039699E-02-6.68410750E+00 6.80837034E+02-1
10-2.48597019E-02 8.23034310E+01-1.04050162E-04 4.10035098E-02-4.00432153E+00 1
215.        2
 8 4.84836491E-01-3.20811892E+01 2.37474528E-01-9.79004947E+01 1.00947849E+04 1
 9 1.76756712E+01-3.94110534E+03 3.17849965E+02-1.43598138E+05 1.62259851E+07 1
218.        3
 1-7.00000000E-02 9.20500000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 8-7.00000000E-02 9.20500000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
10-7.00000000E-02 9.20500000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
290.        1
 1-7.00000000E-02 9.20500000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
```

```
* FUSELAGE + WINGS BODY DESCRIPTION DATA *

    4
 1 1.10000000E+02 1.90000000E+02
22 4.33296327E-02 1.80002653E+01-1.08547881E-02 3.26821039E+00-1.25005203E+02-1
 2 1.90000000E+02 2.10000000E+02
22 8.53087902E-02-1.78558350E-01 0.0          E     0.0          E     1.00000000E+00 0
 3 2.10000000E+02 2.85000000E+02
22 8.95633362E-02 8.81449228E+00 1.60979977E-02-6.67703379E+00 7.89998489E+02-1
 4 2.85000000E+02 4.30000000E+02
22-3.69162194E-02 5.93919575E-01-1.95519518E-02 1.33830055E+01-1.30691653E+03 1
-26.678576      7    1
 1 4.67350412E-01 4.26126281E+01 1.77600524E-01 2.88372638E-01 2.16548185E+02-1
 2 0.0          E+00 1.10150000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 3 0.0          E+00 1.10150000E+01-1.0          E+00-5.22100000E+01-6.80471025E+02-1
 4 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
 5 0.0          E+00 0.0          E+00-1.0          E+00-5.22100000E+01-6.80471025E+02 1
 6 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
 7 0.0          E+00 1.10150000E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
-26.511737      1
 5 6.44442767E-01 2.44436641E+01 1.46386781E-01 7.09346363E+00 1.26419987E+02-1
  -6.105        1
 3-2.48647190E-01-3.99438475E+01 3.22172520E-01-1.52592977E-03 1.66169704E+03 1
   0.           2
 2 1.27651029E-01 1.17943095E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 7 1.27651029E-01 1.17943095E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
  35.           1
 5 2.25972698E-01 1.88438923E+01 3.82148943E-02-1.16294999E+00 3.18535228E+01-1
  50.           1
 3 3.26131567E-02-5.61306675E+00 7.12285310E-03-5.75019262E-01 1.27728399E+01 1
  60.          12   2
 1 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
 2 9.41050000E-02 1.34716110E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 3 3.26131567E-02-5.61306675E+00 7.12285310E-03-5.75019262E-01 1.27728399E+01 1
 5 2.43645384E-01 1.97395403E+01 4.84059774E-02-1.25659158E+00 4.38354577E+01-1
 6 5.61007015E-01-1.87201403E+01-3.62840714E-01 4.67526985E+01-1.34348031E+03-1
 7 9.41050000E-02 1.34716110E+01 0.0          E+00 0.0          E+00 1.0          E+00 0
 8 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
 9 1.03303975E+00-2.69645401E+02 1.48693730E+00-6.81728634E+02 9.69296900E+04 1
10 3.25001860E-01 2.12498820E+01 4.56257987E-02-5.58759941E+00 1.71565599E+02 1
11 4.60000000E-01-3.00000000E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
12 9.25000000E-02 7.77500000E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
13 0.0          E+00 0.0          E+00 0.0          E+00 0.0          E+00 1.0          E+00 0
```

```
 65.           4
 1 2.50000000E-01 2.65000000E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
 8 2.50000000E-01 2.65000000E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
10 3.30800117E-01 2.13816842E+01 2.10071510E-02-2.56454008E+00 7.83197281E+01 1
13 1.69428355E-01-7.61103902E+00 3.62521514E-02-5.18344403E+00 1.87025196E+02-1
 70.           4
 1 4.49999988E-01 1.12031262E+01 1.83749962E-01-2.48062449E+01 8.41337958E+02 1
 3 7.47301953E-02-4.28907043E+00 5.79379569E-03-8.65816229E-01 3.47521388E+01-1
 6 7.62467575E-01-5.13706401E+01-6.28039714E-02 5.64312348E+00-5.50767734E+01 1
10 4.65789000E-01 1.32237330E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
 77.           3
 1 3.19469309E-01 2.20894279E+01-3.44198713E-02 5.18085882E+00-1.93702750E+02 1
 5 7.24999081E-02 1.70083294E+01 1.12293182E-03-2.02127204E-01 9.31348979E+00 1
 8 1.81818203E-01 3.21818144E+01 1.03305539E-02-1.57024414E+00 6.01238176E+01-1
 80.           2
 1 2.63562000E-01 2.74657260E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
 6 5.37634000E-03 9.58602200E+00 0.0           E+00 0.0           E+00 1.0           E+00 0
 85.           2
 2 5.36480156E-02 1.54704910E+01-2.71605902E-03 5.34714063E-01-2.38625259E+01 1
 7-4.02667000E-01 5.32133600E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
 95.36         2
 8 1.29323524E-01 3.31659571E+01 8.33241401E-03-1.55383488E+00 7.73570984E+01 1
13-4.61624110E-01 5.15383731E+01 2.62234164E-01-5.23353921E+01 2.62443194E+03-1
100.           1
 7-4.02667000E-01 5.32133600E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
103.           1
11 2.94177736E-01 2.02578527E+01 5.59112031E-02-1.33961752E+01 8.25069479E+02-1
110.           2
10 3.72282544E-01 2.47202489E+01 1.23135674E-02-2.88544023E+00 1.70045253E+02-1
13 0.0           E+00 0.0           E+00 0.0           E+00 0.0           E+00 1.0           E+00 0
120.           9    4    1.9
 5 4.22221093E-02 2.08296406E+01-7.26924572E-04 1.97586566E-01-1.28325043E+01 1
 7 0.0           E+00 8.92000000E+00 0.0           E+00 0.0           E+00 1.0           E+00 0
12 3.11818358E-02 1.37379328E+01-2.04450978E-03 5.45701343E-01-3.46769581E+01 1
14 5.54759061E-02 2.52983153E+00 3.46164047E-03-1.20290707E+00 1.06256968E+02 1
15-5.54759235E-02 1.96901731E+01 3.46164330E-03-1.20290821E+00 1.06257085E+02-1
16 4.28614138E-02 4.31462046E+00 2.06422324E-03-7.18208885E-01 6.35161184E+01 1
17-4.28614040E-02 1.79053759E+01 2.06422148E-03-7.18208281E-01 6.35160620E+01-1
18 0.0           E+00 1.11100000E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
21 2.51831244E-01-9.54562517E+00 2.33959273E-02-7.46835782E+00 5.96862302E+02 1
122.           1         1.9
 2 0.0           E+00 2.30000000E+01 0.0           E+00 0.0           E+00 1.0           E+00 0
130.           1         1.9
```

```
10 1.44921931E-01 4.55285164E+01-3.31860250E-02 1.01251470E+01-7.07791365E+02 1
 140.        2         1.9
 8 2.05479000E-01 2.72877300E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 9 2.26027000E-01 1.81164900E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 150.        1         1.9
11 1.26667000E-01 3.69666200E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 160.        2         1.9
 1 3.58105141E-01 2.40204689E+01 5.91180929E-02-1.57053527E+01 1.14091266E+03-1
12-4.35421248E-02 2.43167648E+01-1.46009950E-03 5.87340493E-01-5.23091400E+01 1
 170.        1         1.9
 5 0.0        E+00 2.80000000E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 180.        3         1.9
 6 3.03462490E-02 4.93794747E+00 3.98463449E-03-1.37491097E+00 1.18741491E+02 1
 7-5.54759235E-02 1.96901731E+01 3.46164330E-03-1.20290821E+00 1.06257085E+02-1
11 2.35939575E-02 4.64431781E+01-6.09878510E-02 2.23944900E+01-1.97977872E+03 1
 190.        4         1.9
 2 5.01194177E-02 1.30713767E+01 4.53565465E-03-1.72376514E+00 1.64145395E+02 1
 3 0.0        E+00 1.00000000E+00 0.0        E+00 0.0        E+00 1.0        E+00 0
 6 8.00000000E-02-3.40000000E+00 0.0        E+00 0.0        E+00 1.0        E+00 0
10-2.48597019E-02 8.23034310E+01-1.04050162E-04 4.10035098E-02-4.00432153E+00 1
 200.        1         1.9
 1 8.00000602E-02 6.23787330E+01 1.65039699E-02-6.68410750E+00 6.80837034E+02-1
 203.        6         1.9
 2-6.76794001E-02 2.26207604E+01-1.01106588E-02 9.05011740E+00-1.18313633E+03 1
 7 6.77003943E-02 7.93805583E+00-1.01105037E-02 9.05169351E+00-1.18336783E+03-1
14-6.76859833E-02 1.42810702E+01-1.01106792E-02 9.05065979E+00-1.18321397E+03 1
15 6.77003943E-02 7.93805583E+00-1.01105037E-02 9.05169351E+00-1.18336783E+03-1
16 9.27609874E-01-8.52069135E+01 9.09851308E-01-2.17906746E+02 1.45601058E+04-1
17-3.12060699E-01 4.33002563E+01 1.12311500E-01-3.22998628E+01 2.69014825E+03 1
 212.        2         1.9
 8 9.63279045E-01-1.26160629E+02-5.64020269E-02 2.34827545E+01-2.44257980E+03-1
 9 1.76756712E+01-3.94110534E+03 3.17849965E+02-1.43598138E+05 1.62259851E+07 1
 215.        1         1.9
 8-2.48597019E-02 8.23034310E+01-1.04050162E-04 4.10035098E-02-4.00432153E+00 1
 218.        3         1.9
 1-7.00000000E-02 9.20500000E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 8-7.00000000E-02 9.20500000E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
10-7.00000000E-02 9.20500000E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 220.        1    3    1.9
 1-7.00000000E-02 9.20500000E+01 0.0        E+00 0.0        E+00 1.0        E+00 0
 230.        3
19 3.36342972E-01-3.08786588E+01 7.70106182E-03-3.64537091E+00 4.31450575E+02 1
20 3.36342972E-01-3.08786588E+01 7.70106182E-03-3.64537091E+00 4.31450575E+02 1
```

```
21 4.04026000E-01-3.37448100E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 240.        2
 1-3.31343384E-01 1.54469238E+02 7.63460300E-02-3.63267175E+01 4.32177736E+03-1
 6 3.78260866E-02 5.83620332E+04 2.33422784E+02-1.12296658E+05 3.41887176E+09-1
 250.        3
 1-3.02083331E-01 1.44458333E+02 1.32920989E-01-6.46102344E+01 7.85437568E+03 1
19 4.04026000E-01-4.67448100E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
20 4.04026000E-01-4.67448100E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 286.41      2
 1 0.0         E+00 7.20000000E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 6 0.0         E+00 1.58000000E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 289.96      4
 2 0.0         E+00 2.73100000E+01 0.0         E+00 0.0         E+00 1.0         E+00 0
 7-4.15609201E+00-7.56278893E+03 2.35265344E+01 5.93082172E+04 5.77579183E+07 1
14 4.14592346E+00 7.56669858E+03 2.34269558E+01 5.90112026E+04 5.74799363E+07-1
15-4.15609201E+00-7.56278893E+03 2.35265344E+01 5.93082172E+04 5.77579183E+07 1
 480.        2
16-9.55760562E-02 1.87374283E+02 5.77347121E-02-6.07877908E+01 3.31783010E+04-1
17 4.61274458E-02-4.67336905E+01 1.47273267E-02-1.21181837E+01 3.77934000E+03 1
```